

Publicación práctica para usuarios de

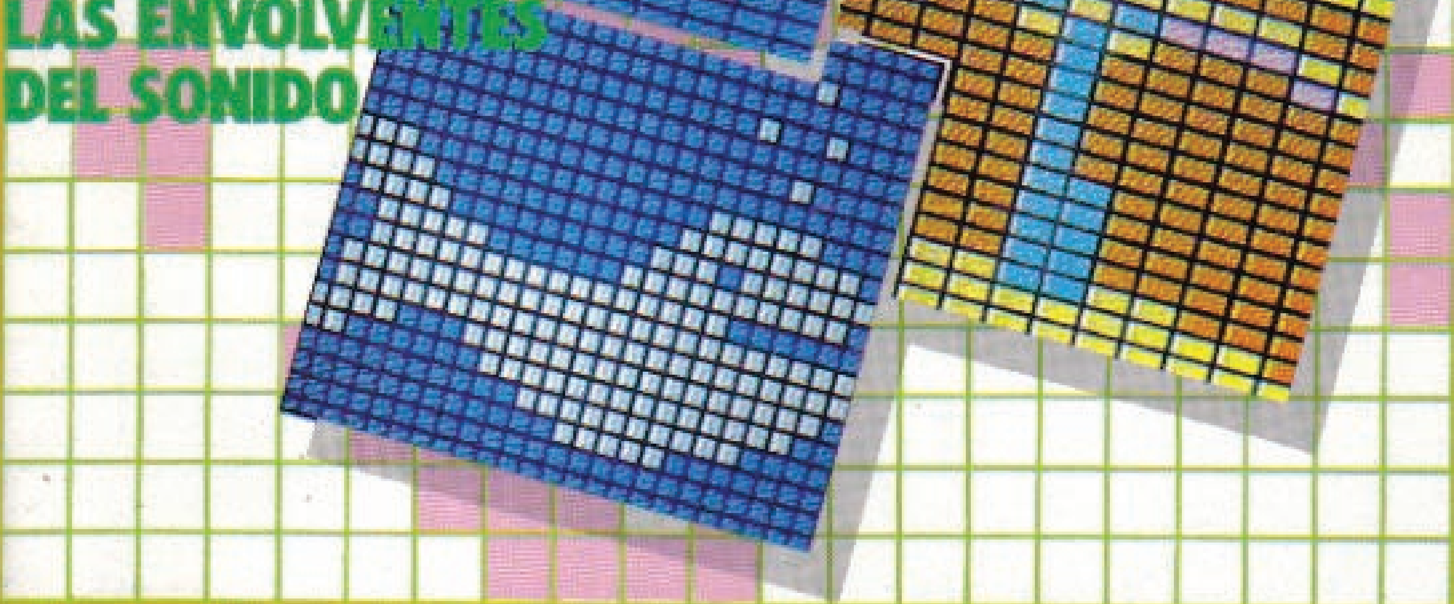
commodore

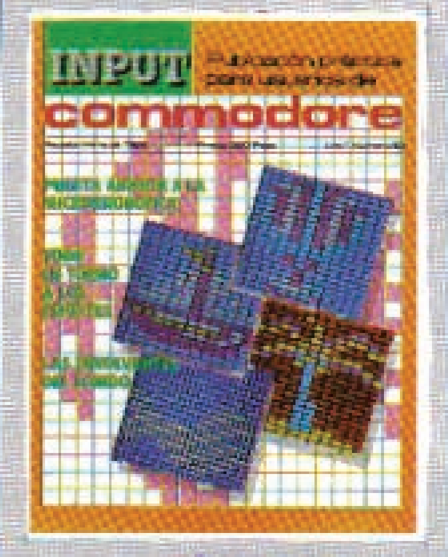
Revista mensual 1986 Precio 350 Ptas Año 1 Número 10

PUERTA ABIERTA A LA MICRORROBOTICA

TODO EN TORNO A LOS ESPRITES

LAS ENVOLVENTES DEL SONIDO





AÑO 1 NUMERO 10

DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Luis R. Palencia, Christophe Aris, Francisco Tórtola, Benito Román, Esther de la Cal

INPUT Commodore es una publicación juvenil de EDICIONES FORUM

GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: Grupo Jota
Madrid: c/ General Varela, 35
Telef. 270 47 02/03
Barcelona: Avda. de Sarriá, 11-13, 1.º
Telef. 250 23 99

FOTOMECANICA: Ochoa, S. A.

COMPOSICION: EDCA, S. A.

IMPRESION: Sirven Grafic
C/ Gran Vía, 754-756, 08013 Barcelona
Depósito legal: M. 27.884-1985

SUSCRIPCIONES: EDISA,
López de Hoyos, 141, 28002 Madrid
Telef. (91) 415 97 12

REDACCION:
Alberto Alcocer, 46, 4.º
28016 Madrid. Telef. 250 10 00

DISTRIBUIDORA:
SGEL, PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Cobarsí
08006 Barcelona

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

INPUT commodore

SUMARIO

MAS AMIGA QUE NUNCA

Durante la primera quincena del pasado mes de mayo se celebró en Londres el tradicional **Commodore Show,** la más importante feria europea para usuarios de esta marca. Allí estuvo INPUT. La impresión de lo visto es de enorme contraste con respecto a ediciones anteriores. Fundamentalmente, aquello parecía más un mercadillo de liquidación de programas que una feria de presentación de nuevos productos. Las principales casas de *software* no disponían de *stand* propio. Trascendía una sensación de época transitoria para **Commodore.** También es cierto que aquel mercado tiene una mayor saturación en número de **64** vendidos.

En aquel maremagnum brillaba con luz propia el fabuloso **Amiga.** Aunque no había una gran cantidad de *software* de aplicaciones, los programas gráficos de **Electronic Arts,** el sorprendente digitalizador de vídeo de la firma alemana **Fischer Technik** o la base de datos de **Precision Software,** fueron suficiente para ofrecernos una idea de las enormes posibilidades potenciales aportadas por el codiciado **Amiga.** No obstante, no es preciso ir tan lejos para verlo funcionar. También fue presentado en la feria **Informat,** de Barcelona, para el mercado nacional.

A pesar de que los programadores profesionales argumentan una gran complejidad de programación, las capacidades gráficas del **Amiga** son sencillamente sorprendentes, demostrándose que es más importante la variedad de la paleta de colores que incluso la resolución. El *chip* de vídeo, unido a la velocidad de proceso del microprocesador **68000,** produce imágenes que bien pueden ser consideradas dibujos animados.

Cambiando de terreno, debemos reconocer que la picaresca y las ganas de broma siguen siendo uno de los rasgos característicos de nuestro país. Pues bien, hemos recibido algunas llamadas de lectores que habéis recibido una carta (eso sí, llena de tachaduras) en las que se comunica la concesión de un hipotético premio. Los premiados en INPUT aparecen únicamente en las listas publicadas en la revista y reciben posteriormente una carta convenientemente firmada, que incorpora el logotipo de la editorial (no el de la revista). Así que, atentos a las bromitas.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Alberto Alcocer, 46 - 4.° B. 28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido

Segundo título elegido

Tercer título elegido

Programa que te gustaría conseguir

Qué ordenador tienes

Nombre

1.° Apellido

2.° Apellido

Fecha de nacimiento

Teléfono

Dirección

Localidad

Provincia

AHORA TE PUEDES PROGRAMAR UN VIAJE ALUCINANTE AL EPCOT CENTER CON ALEA. ES LOGICO.

Te presentamos, en estreno mundial, todo un reto a tu inteligencia: la colección de programas Logicolor.

Con los tres juegos de la colección Logicolor tu mente desafiará a la fría lógica del ordenador. ¡Atrévete con ellos!

AUTOS LOCOS: Construye tu propia escudería y apuesta por tu bólido favorito. Un primer contacto con el uso de los símbolos. Para chicos entre 10 y 12 años. Incluye también un super-master mind contra el ordenador.

MANZANAS Y GUSANOS: Utilizando fórmulas puedes recoger las manzanas y dejar fuera los gusanos, proteger las ánforas de los golpes del martillo, defender los globos aerostáticos de las flechas enemigas, o evitar que los cañones destruyan las torres de tu fortaleza. Tu inteligencia lógica es la única arma que necesitas. Para chicos entre 12 y 14 años, y para quienes desean mantener su mente en forma.

REHENES: Tendrás que desarrollar una estrategia lógica si quieres eliminar a los conspiradores y salvar la corona. ¿Te gustaría descubrir la fórmula que abre el cofre de los diamantes? Intenta descubrir un procedimiento lógico para rescatar a los rehenes. Para chicos entre 14 y 16 años, y para quienes se las dan de genios.

Además, la compra de cada programa de la colección Logicolor te da derecho a participar en el fabuloso concurso EPCOT, y si consigues vencer al ordenador, tus posibilidades de conseguir un magnífico premio se duplican.

Si resultas ganador puedes elegir uno de estos SUPERPREMIOS:

1) UN FANTASTICO VIAJE PARA DOS PERSONAS DE 9 DIAS AL EPCOT CENTER: visitarás Marineland, el Museo Aeroespacial de la Nasa, Disneyworld, el Epcot Center y otros muchos lugares.
2) UN SUPERORDENADOR IBM-PC portátil.
3) UNA PAGA MENSUAL DE 30.000 Pts. durante un año para ti solo.

Encontrarás las Bases para participar en el concurso, junto con las fichas, en cada programa. Envíalas a ALEA antes del 2[illegible] de Julio de 1986.

¡Atención!, si envías tus fichas antes del 23 de Junio, tus posibilidades de ganar son aún mayores.

Alea también ha pensado en los más "peques", tus hermanos de 4 a 8 años. Para ellos tenemos una serie de juegos que les ayudarán al aprendizaje de la escritura y la lectura. Comprando cualquiera de ellos, participarás automáticamente en el concurso LEXA, pudiendo llegar a conseguir una beca de estudios de hasta 500.000 Pts.

Puedes pedir tus programas llamando al teléfono de Madrid (91) 446 57 64 o bien enviándonos el cupón que hay al pie de esta página. También encontrarás los programas de la colección Logicolor en la microtienda de tu barrio, El Corte Inglés y Galerías Preciados.

Ánimo, por sólo 3.875 Pts. obtienes un magnífico programa y ¡hasta cuatro participaciones para el gran concurso EPCOT!

¡NO LO DEJES ESCAPAR! Prográmate ahora mismo un premio alucinante.

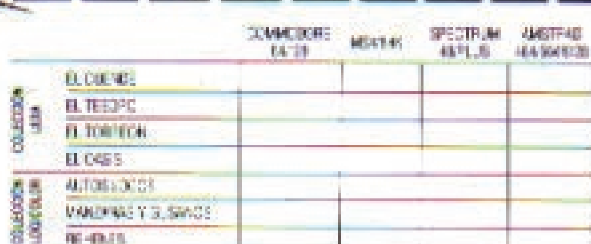


apartado de correos 10.048 de Madrid

GRUPO SOFT

TCA

INPUT COMMODORE



		COMMODORE 64/128	MSX 64K	SPECTRUM 48/PLUS	AMSTRAD 464/664/6128
COLECCION LEXA	EL CUENTO				
	EL TEBEO				
	EL TORREON				
	EL OASIS				
COLECCION LOGICOLOR	AUTOS LOCOS				
	MANZANAS Y GUSANOS				
	REHENES				

[illegible]

Forma de pago:
- Contra reembolso
- Con cheque adjunto a nombre de Alea, S.A.
- Con cargo a mi tarjeta de crédito
  - Visa
  - American Express
  - Diners Club

Nombre del titular: ____
N.º de la tarjeta: ____
Válida desde ____ hasta ____
Firma del titular
Nombre: ____
Dirección: ____
Tfno: ____



THE FINAL CARTRIDGE AMPLIADO

El pasado mes dedicabamos un articulo a la descripcion del cartucho de utilidades The Final Cartridge. Pues bien, con posterioridad a su elaboracion acaba de aparecer una nueva version ampliada, que ofrece algunas interesantes ventajas; como por ejemplo el freeze, capaz de "congelar" la memoria del C-64 para su posterior lectura. No obstante, estamos trabajando en un nuevo articulo que el mes que viene dará buena cuenta de las nuevas aportaciones.



COMPULOGICAL, NUEVAS REPRESENTACIONES

La firma distribuidora de software Compulogical ha firmado nuevos contratos de representacion exclusiva con el fabricante Softech/Edoe para el lanzamiento aqui de sus nuevos productos. El primer programa en ver la luz será Fireline en versiones para cassette y disco.
El contrato con Consolidated Software aporta a su catalogo el titulo que tiene como protagonista a la bruja Fiona.
Por otro lado, de su tradicional representado Epyx lanzara varios titulos. The movie monster games es una historia basada en monstruos de pelicula. Vorpal proporciona utilidades para el C-64, entre las que se encuentra un cargador rapido que se puede incorporar a programas propios en BASIC. Supercicles es el ultimo titulo por el momento en el que una moto compite por una carretera del estilo Pole Position.



PROGRAMAS EN TARJETA

Una de las posibles alternativas que se les presentan a los fabricantes y distribuidores de software es la implementacion de sus programas en memoria ROM. Para los Commodore se viene utilizando el clasico cartucho y ahora le llega el turno a las tarjetas planas ya empleadas en los MSX, que vienen a ser lo mismo pero mas comodo de manejar. La tarjeta Vicard está a punto dar el salto en nuestro pais. Su formato, similar al de una tarjeta de crédito, precisa un adaptador para la conexion al port del usuario.



DROSOFT/ARIOLA

Esta primavera viene plagada de firmas. Esta vez es Drosoft quien ha llegado a un acuerdo con la britanica Ariolasoft, distribuidora de los productos de Electronic Arts en aquel pais.
Drosoft tiene en sus manos el increible programa deportivo Golf Construction Set, donde aparece un jugador que utiliza los palos con un realismo increible (ver softactualidad de este mes), siendo desarrollado por el equipo Hungarian Andromeda.



AMIGA, AL FIN...

El esperado Amiga acaba de hacer su entrada triunfal en Europa. La presentación oficial a la prensa nos fue hecha el pasado 9 de mayo en el Commodore Show celebrado en Londres.
Unos dias despues ha visto la luz en el Informat de Barcelona.
Una cosa queda clara: Es una máquina sensacionalmente exquisita.
Confiamos disponer en breve de un modelo que describir a nuestros lectores. Los leones de redaccion rugen hambrientos.

CORRECCION

En el pasado numero de INPUT Commodore nos haciamos eco del concurso convocado por Erbe Software para recibir el *software* editado por la firma durante un año. Pues bien, los clasicos duendes que enredan en las redacciones nos jugaron la mala pasada de introducir alguna errata. Por lo pronto, la fecha tope de recepcion sera el proximo 30 de agosto. Los cupones a incluir son como se indica a continuacion, ademas de los logotipos originales citados en la convocatoria. Lamentamos el error al tiempo que os animamos a participar.


Mayo


Junio

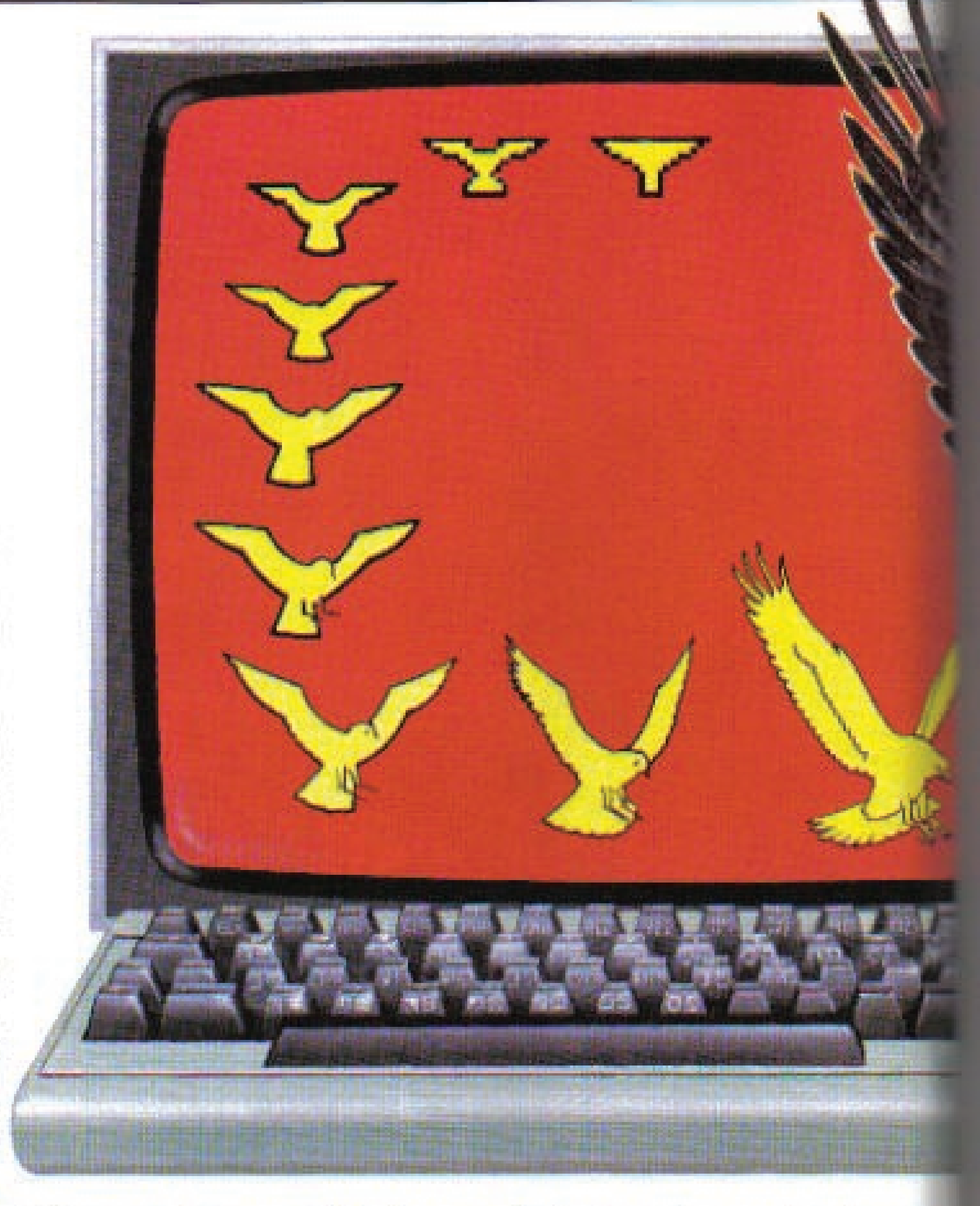

Julio


Extra verano

Junio

COMO MANEJAR LOS SPRITES EN EL COMMODORE 64

El sprite es una utilidad gráfica estandar del Commodore 64 **que produce una imagen de alta resolución y gran movilidad. Es muy sencillo de manejar y constituye la base de muchos juegos.**



Un *sprite*, también llamado un bloque objeto móvil, es una clase especial de GDU (Gráfico Definido por el Usuario) de alta resolución y gran movilidad, algunos de cuyos usos ya hemos visto en nuestra revista. Aparte de su movilidad, tiene otras características poco comunes, como son su capacidad de extenderse y contraerse en anchura y en altura mediante un comando. Por ello no es de extrañar que se encuentren con tanta frecuencia los *sprites* en la programación de juegos; de hecho se pueden utilizar en cualquier programa en el que se requieran gráficos animados de alta resolución. Por ejemplo, un programa de una aplicación comercial podría utilizar uno o más *sprites* para formar un símbolo que represente la opción elegida del programa, o como parte de un logotipo animado.

Existen dos tipos de *sprites*: los de forma estandar de alta resolución y los multicolor. La diferencia entre ambos consiste en que los primeros adoptan el color que se haya elegido para la visualización en pantalla, mientras que los multicolor permiten una elección de hasta cuatro colores simultáneos, aunque a costa de sacrificar un poco de resolución horizontal.

Para empezar, fijemos nuestra atención en los *sprites* estandar. Los multicolor se tratarán separadamente.

DEFINIENDO UN SPRITE

Los *sprites* se definen de una forma muy parecida a los GDU's. No obstante el *sprite* es más largo y requiere más sentencias DATA para definirlo. De hecho puede tener el triple de anchura de un GDU y casi tres veces su altura, ocupando un área de 24 *pixels* por 21. No tienen que utilizarse necesariamente las 504 posibles posiciones para un *pixel* (es decir, no hay por qué activarlas todas ellas) y puede que en algunos casos no sea posible hacerlo.

En vez de definir separadamente los 24 *pixels* de cada fila, la información sobre la forma de un *sprite* está contenida en cada una de las 21 líneas en tres grupos de ocho (tres bytes). Esto nos da una clave sobre la forma en que los valores de las sentencias DATA utilizados en los GDU's normales se pueden adaptar para ser utilizados en este caso, ya que dichos valores también están calculados en grupos de ocho.



La situación de los bytes en el *sprite* adopta la forma que muestra la tabla 1:

```
Fila 1: BYTE1   BYTE2   BYTE3
Fila 2: BYTE4   BYTE5   BYTE6
Fila 3: BYTE7   BYTE8   BYTE9
```

... y así continúa hasta:

```
Fil.20: BYTE58 BYTE59 BYTE60
Fil.21: BYTE61 BYTE62 BYTE63
```

De esta forma tienes los grupos de tres bytes para cada una de las 21 filas del *sprite*.

Cada uno de estos bytes maneja la información de la misma manera que una de las líneas individuales de los GDU's. Por ejemplo, considerando la notación decimal, los posibles valores de las posiciones de los *pixels* a lo largo de cada línea son los que se muestran en la figura 1.

Según esto, si se utilizan las ocho posibles posiciones de un *pixel* dentro de un byte (es decir los ocho bits), estando pues todas ellas activadas, el correspondiente valor decimal de dicho byte será 128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255. Si no se utilizan *pixels* en un byte, la suma de los valores de los bits es 0. Los valores comprendidos entre estos dos extremos se corresponden unívocamente con todas las combinaciones posibles de los *pixels* «usados». El valor obtenido de esta manera se puede utilizar directamente en una sentencia DATA como parte de la definición de un *sprite*.

PRIMEROS PASOS

Veamos ahora algunos ejemplos. Igual que en el caso de los GDU's ordinarios, la mejor manera de empezar es dibujar sobre un papel cuadriculado la forma deseada. Marca primero en dicho papel una zona que contenga 24 por 21 cuadrados, y señala las divisiones de los bytes. También puedes fotocopiar o calcar la retícula del *sprite* que presentamos aquí o la contenida en el manual. Como ejemplo presentamos un *sprite* que forma parte de un juego con estación espacial.

En dicho ejemplo puedes apreciar los correspondientes valores de las sentencias DATA. Fijémonos en ellos un poco más de cerca. El primer byte de la primera fila no se utiliza, por lo que el correspondiente valor DATA de estos *pixels* es 0. En el siguiente byte se activan todos los *pixels* menos uno, el correspondiente al último lugar. El valor total correspondiente es pues 128 + 64 + 32 + 16 + 8 + 4 + 2 = 254. El tercer byte de la primera fila tampoco se utiliza, correspondiéndole otro 0 en la sentencia DATA. En consecuencia los valores de la fila 1 son 0, 8, 0.

Antes de seguir adelante, echemos una ojeada a la organización de los *pixels* en el resto del *sprite*. Muchas veces hay otra fila que utiliza en gran parte la misma configuración para uno o más bytes, especialmente en los dibujos simétricos. En estos casos te puedes ahorrar el tener que calcular los valores de las sentencias DATA más de una vez. No podrás hacer esto en la fila 1, pero sí podrás en las filas 14 y 16 o en las filas 17 y 19.

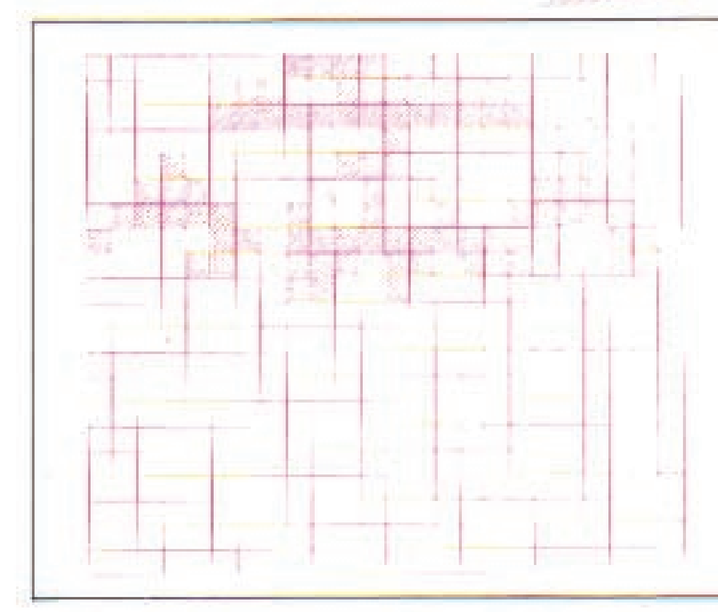
Veamos ahora la siguiente fila. En el primer byte, los *pixels* 2 y 1 están activados, con lo que el correspondiente valor es 2 + 1 = 3. En el siguiente byte están activados los *pixels* 6, 5, 4 y 1, por lo que el valor será 32 + 16 + 8 + 1 = 57. Finalmente en el último byte está activado únicamente el primero (128). El valor total de la fila es pues 3, 57, 128.

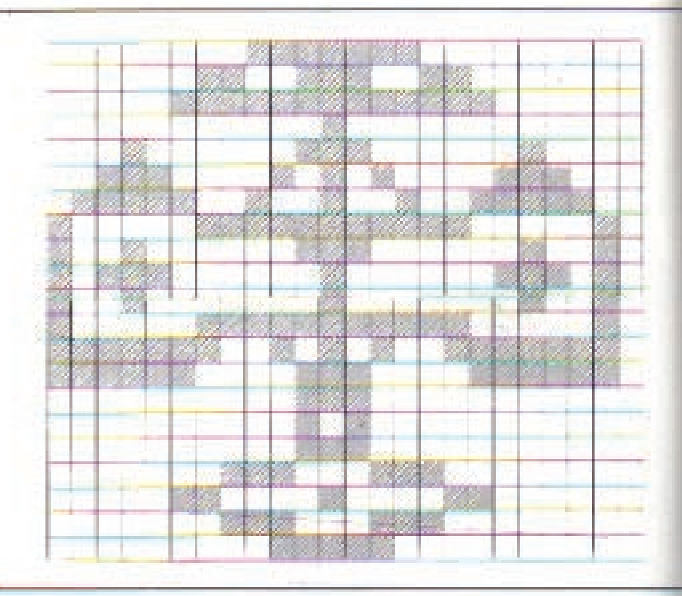
El primer byte de la fila 3 tiene activados tres *pixels*, con un total de 4 + 2 + 1 = 7. En el segundo byte están activados todos, dando un valor máximo de 255. El valor del último byte es 128 + 64 = 192. Comprueba de la misma forma los valores de las restantes filas.

Una vez hecho esto, puedes agrupar todos los valores de las filas para obtener una sentencia DATA. En el caso del *sprite* de la estación espacial, las sentencias son las siguientes:



```
20 DATA 0,254,0,3,57,128,7,
   255,192,0,16,0,16,56,16,
   56,84,56,124,148,124,131,
   255
30 DATA 130,144,58,18,184,16
   ,58,144,16,18,131,255,130
   ,254,84,254,252,56,126,0,
   56
40 DATA 0,0,40,0,0,56,0,1,
   199,0,6,16,192,1,199,0,0,
   124,0
```





Así es como se genera el *sprite* en el programa de la estación espacial. Pero no siempre es necesario escribirlo «in extenso» de esta forma. Con los adecuados ajustes en el programa, normalmente es posible omitir las filas no utilizadas del sprite al principio o al final de una definición.

Los restantes ejemplos corresponden a otros dibujos menos complicados en los que el cálculo de los valores es mucho más directo. Intenta seguir la forma en que se obtienen los valores y trata de localizarlos en las sentencias DATA de las líneas 20 a 140 del programa de la estación espacial.

GENERACION DE LOS DATOS

La parte menos divertida del trabajo con el *sprite* es el cálculo de los valores necesarios para definir las 63 sentencias DATA. En vez de hacerlo a mano puedes usar un programa de utilidades que haga el cálculo por tí. Hay en el mercado una gran cantidad de productos de este tipo entre los que puedes elegir.

Pero para empezar lo único que necesitas es un programa relativamente sencillo que te permita «dibujar» el *sprite* deseado sobre la pantalla, modificándolo a tu capricho. Este programa calculará automáticamente los correspondientes valores de las sentencias DATA, que te pueden servir para definir el *sprite* y utilizarlo en otros programas.

Esto es exactamente lo que hace el siguiente programa, dándote además la opción, si tienes una impresora, de sacar una copia en papel que te sirva como referencia en el futuro. El programa sólo vale para *sprites* de alta resolución y un solo color, a pesar de lo cual resulta enormemente útil. Aunque te parezca que teclear este programa puede ser un camino muy largo para hacer los cálculos de un *sprite*, cuando lo hayas hecho una sola vez puedes almacenar el programa (SAVE) y utilizarlo nuevamente cada vez que tengas que definir un sprite:

Teclea

```
10 POKE 53280,1:POKE 53281,1
   :DIM A$(21),Z(3,21),A(24)
20 PRINT"[SHIFT+CLR/HOME]
   [CTRL+5]"TAB(13)"EDITOR
   DE SPRITES[2*CRSR abajo]"
30 PRINT TAB(8);:INPUT
   "[CTRL+7]OPCION IMPRESORA
   .([CTRL+6]S/N[CTRL+7])
   [CTRL+6]";I$:IF I$="S"
   THEN PR$="S"
40 IF I$<>"S" AND I$<>"N"
   THEN GOTO 20
50 PRINT"[SHIFT+CLR/HOME]
   [CTRL+3]"TAB(13)"ESPERA"
60 FORZ=1 TO 8:READ A(Z):A
   (Z+8)=A(Z):A(16+Z)=A(Z):
   NEXT:DATA128,64,32,16,8,
   4,2,1
70 FOR Z=1 TO 21:READ A$(Z)
80 FOR ZZ=1 TO 8:IF MID$(A$
   (Z),ZZ,1)="*"THEN Z(1,Z)=
   Z(1,Z)+A(ZZ)
90 NEXT ZZ:FOR ZZ=9 TO 16:IF
   MID$(A$(Z),ZZ,1)="*" THEN
   Z(2,Z)=Z(2,Z)+A(ZZ)
100 NEXT ZZ:FOR ZZ=17 TO 24:
    IF MID$(A$(Z),ZZ,1)="*"
    THEN Z(3,Z)=Z(3,Z)+A(ZZ)
110 NEXT ZZ,Z:PRINT"[SHIFT+
    CLR/HOME][CRSR abajo]":
    IF PR$="S" THEN OPEN 4,4
    :CMD 4
120 PRINT TAB(17)"[CTRL+3]
    DATOS[CRSR abajo]
    [CTRL+7]"
130 FOR Z=1 TO 21:PRINT Z(1,
    Z);",";Z(2,Z);","Z(3,Z);
    :IF Z<21 THEN PRINT",";
140 NEXT Z:PRINT:PRINT:IF
    PR$="S" THEN GOTO 170
150 PRINTTAB(7)"[CRSR abajo]
    [CTRL+5]PRESIONA
    <RETURN> PARA CONTINUAR"
160 GET K$:IF K$<>CHR$(13)
    THEN GOTO 160
170 PRINT"[SHIFT+CLR/HOME]"
    TAB(15)"[CTRL+3]DISENO
    [CTRL+7][CRSR abajo]"
180 PRINT TAB(7)"[CTRL+5]765
    43210765432107654321O"
190 FOR Z=1 TO 21:PRINT TAB
    (7)"[ctrl+7]";
```


Practica
el mejor deporte con
tu Commodore
TOUR DE FRANCE
ACTIVISION
Entra en el mundo ciclista a través de las 16 etapas del Tour. Con acompañamiento musical y el jersey amarillo esperando al ganador.
C
DECATHLON
Número uno de todas las listas. Y va a quedarse ahí. Va a exigirle una total entrega. Velocidad, fuerza, coraje. El deseo de vencer.
CM
BARRY McGUIGAN
WORLD CHAMPIONSHIP
BOXING
Juego en el que puedes crear a tu propio boxeador. Elige su raza, estilo físico e imagen. Entrénalo y demuestra sus habilidades.
CS
ON-COURT
TENNIS
ACTIVISION
Elige la superficie de la pista y el oponente y demuestra tu control de la raqueta jugando a dobles o individual.
C
Disponibles para:
COMMODORE C
SPECTRUM S
MSX M

```
200 FOR ZZ=1 TO 24:IF MID$
    (A$(Z),ZZ,1)="*"THEN
    PRINT"[CTRL+9][ESPACIO]
    [CTRL+0]";:GOTO 220
210 PRINT".";
220 NEXT ZZ:PRINT"[CTRL+5]";
    Z:NEXT Z:IF PR$="S" THEN
    PRINT#4," ":CLOSE 4
230 IF PR$<>"S" THEN GOTO
    230
240 REM  765432107654321076543210
250 DATA"
260 DATA"
270 DATA"
280 DATA"
290 DATA"
300 DATA"
310 DATA"
320 DATA"
330 DATA"
340 DATA"
350 DATA"
360 DATA"
370 DATA"
380 DATA"
390 DATA"
400 DATA"
410 DATA"
420 DATA"
430 DATA"
440 DATA"
450 DATA"
```

Para utilizar el programa, teclea la instrucción LIST 240-. Aparecerán las 21 filas que hay al final del programa, que son líneas de DATA vacías en las que puedes introducir los datos correspondientes a la forma del *sprite* que quieras dibujar. Cada posición de carácter dentro de la sentencia DATA representa una posible posición de un *pixel*; la línea tiene en total una anchura de 24 caracteres. Para moverte por las líneas, utiliza los controles del cursor, y a continuación no tienes más que poner un asterisco en cada punto de tu dibujo, acordándote de pulsar [RETURN] para introducir las líneas.

Para que te resulte de guía, la sentencia REM de la línea 240 numera las 24 posiciones horizontales disponibles en cada una de las 21 líneas verticales.

En la parte inferior izquierda de esta página te presentamos un dibujo típico. Representa el conjunto de asteriscos utilizados para definir la figura de un pájaro volando.

Cuando hayas completado el dibujo, ejecuta el programa. En la pantalla aparecerá un mensaje preguntándote si deseas una copia en papel. Si tecleas N, la salida se efectuará a través de la pantalla. Primero vienen los 63 números de la sentencia DATA; si no tienes impresora, tendrás que copiarlos a mano. Para el ejemplo mostrado, los datos serían los siguientes

Teclea

```
DATOS 128,0,3,192,0,30,240,0
  ,250,104,1,52,84,2,228,58,
  2,216,45,7,144,20,133,32,
  22,78,64,22,44,64,11,24,
  128,5,201,0,3,230,0,0,49,
  192,16,64
```

Puedes obtener un dibujo a gran escala del *sprite* que has creado pulsando [RETURN].

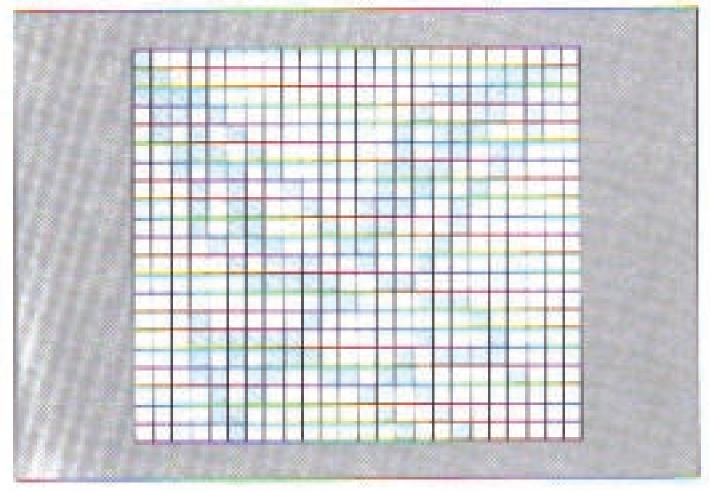
Si ahora quieres tener una copia en papel utilizando la impresora, pulsa al mismo tiempo las teclas [RUN/STOP] y [RESTORE] y ejecuta de nuevo el programa. Cuando te aparezca esta vez la opción de impresora, pulsa S. La salida de impresora empieza con un listado de todos los caracteres de los DATA, seguido de un dibujo a gran escala del *sprite* sobre su retícula base.

Cada uno que definas por medio de este programa puede ser salvado como una parte del mismo. También puedes volver a cargarlo y modificarlo cuando sea necesario para crear nuevos *sprites*.

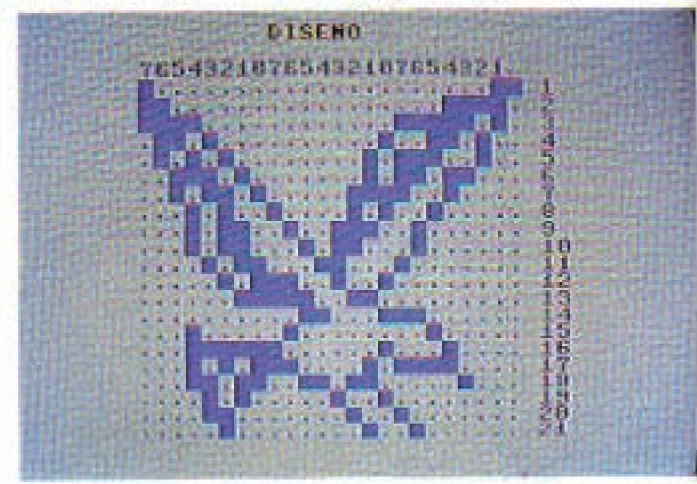
USO DEL SPRITE

La fase de dibujo y definición de los DATA es quizá la parte más fácil del uso de un *sprite*. Cualquiera que sea la forma en que obtienes los valores de tus DATA, en sí mismos no te sirven para nada. Sólo puedes hacer uso del sprite que acabas de crear cuando incorporas dichos valores en un nuevo programa.

Primero tienes que almacenar la información del *sprite* en algún sitio conveniente dentro de la memoria de tu ordenador. A continuación puedes lla-



En primer lugar tienes que dibujar tu pájaro volador sobre papel cuadriculado. A continuación lo puedes introducir a base de asteriscos en el programa generador de datos.



marlo y manipularlo dentro de tu programa.

Más adelante nos ocuparemos de la forma exacta en que puedes hacer esto. Por el momento teclea y ejecuta el siguiente programa:

Teclea

```
10 V=53248:X=150:Y=157:PRINT
   "[SHIFT+CLR/HOME]"
20 FOR I=16000 TO 16062:READ
   A:POKE I,A:NEXT I
25 POKE 2040,250:POKE V+21,1
   :GOTO 50
30 GET A$:A=0:XX=0:IF A$="P"
   THEN A=1:GOTO 50
35 IF A$="L" THEN A=2:GOTO50
40 IF A$="Z" THEN XX=-2
45 IF A$="X" THEN XX=+2
50 FOR Z=1 TO 10:X=X+XX:IF X
   >250 THEN X=30
55 IF X<20 THEN X=250
60 IF A=1 AND Y>70 THEN Y=
   Y-2
65 IF A=2 AND Y<200 THEN
   Y=Y+2
70 POKE V,X:POKE V+1,Y
75 NEXT Z:GOTO 30
100 DATA 128,0,3,192,0,30,
    240,0,250,104,1,52,84,2,
    228,58,2,216,45,7,144,20
105 DATA 133,32,22,78,64,22,
    44,64,11,24,128,5,201,0,
    3,230,0,0,49,192,16,64
110 DATA 32,31,129,16,21,194
    ,240,27,54,8,10,9,0,12,4
    ,128,4,2,64
```

Este programa hace que el ordenador defina un *sprite* y desplace su imagen sobre la pantalla bajo control del teclado utilizando las teclas L, P, Z y X. El *sprite* elegido en este caso es la figura del pájaro volador definida anteriormente, pero puedes usar este mismo programa para cualquier otro *sprite* de un solo color que te inventes; para ello lo único que tienes que modificar son las sentencias DATA del final.

Observa las líneas 100, 105 y 110 y verás que en ellas se listan los valores de los datos del pájaro volador que fueron generados por el programa de utilidad.

En la línea 10 se establecen una se-

Decimal	V+valor	Descripcion
53248	V	Posicion X sprite-0
53249	V+1	Posicion Y sprite-0
53250	V+2	Posicion X sprite-1
53251	V+3	Posicion Y sprite-1
53252	V+4	Posicion X sprite-2
53253	V+5	Posicion Y sprite-2
53254	V+6	Posicion X sprite-3
53255	V+7	Posicion Y sprite-3
53256	V+8	Posicion X sprite-4
53257	V+9	Posicion Y sprite-4
53258	V+10	Posicion X sprite-5
53259	V+11	Posicion Y sprite-5
53260	V+12	Posicion X sprite-6
53261	V+13	Posicion Y sprite-6
53262	V+14	Posicion X sprite-7
53263	V+15	Posicion Y sprite-7
53264	V+16	Bit mas significativo de la coordenada X
53265	V+17	Regsitro de control del VIC
53266	V+18	Registro de barrido
53267	V+19	(Lapiz optico)
53268	V+20	(Lapiz optico)
53269	V+21	Activacion de sprites
53270	V+22	Registro de control del VIC
53271	V+23	Expansion en Y de los sprites 0 a 7
53272	V+24	Control de memoria del VIC
53273	V+25	Registro de interrupcion
53274	V+26	Activacion de interrupcion
53275	V+27	Prioridad del fondo
53276	V+28	Seleccion multicolor
53277	V+29	Expansion en X de los sprites 0 a 7
53278	V+30	Colision de sprites
53279	V+31	Colision con fondo
53280	V+32	Color del borde de la pantalla
53281	V+33	Color de fondo 0
53282	V+34	Color de fondo 1
53283	V+35	Color de fondo 2
53284	V+34	Color de fondo 3
53285	V+37	Multicolor 1 de Sprites
53286	V+38	Multicolor 2 de sprites
53287	V+39	Color sprite 0
53288	V+40	Color sprite 1
53289	V+41	Color sprite 2
53290	V+42	Color sprite 3
53291	V+43	Color sprite 4
53292	V+44	Color sprite 5
53293	V+45	Color sprite 6
53294	V+46	Color sprite 7

ne de variables —fundamentalmente un punto de partida para el *sprite*— y se borra la pantalla. La línea 20 lee los datos que definen el *sprite* y los mantiene en la memoria del ordenador. La línea 30 hace que aparezca la imagen. Las líneas 30 a 55 informan al ordenador sobre la forma en que debe moverse el *sprite* bajo el control del teclado. Si quieres mover el *sprite*, tendrás que editar las líneas 40, 50, 55, 60, 70 y 75 del programa. En este caso el pájaro aparecerá en un punto fijado por las variables X e Y, cuyos valores son 150 y 157, en las proximidades del centro de la pantalla. Si en la línea 10 introduces valores distintos para X e Y, puedes hacer que el pájaro volador aparezca en el sitio que tú quieras.

Si estás utilizando el programa de números del *sprite*, aquí tienes una forma rápida de transferir un dibujo a las líneas de las sentencias DATA del programa. Usando papel de calcar, marca las líneas clave de tu diseño, que debe ser compatible con el tamaño de la pantalla, y seguidamente pégalo en la pantalla del televisor después de que hayas listado las líneas 240 a 450. Ahora ya no tienes más que utilizar el cursor y dibujar asteriscos «debajo» de las líneas de tu dibujo hasta que el conjunto de asteriscos coincida lo mejor posible con el dibujo del papel.

Probablemente la parte más complicada es la que se extiende desde las líneas 10 a 25, que es la parte que realmente almacena el *sprite* en la memoria del ordenador. Pero no te preocupes si no la entiendes completamente por ahora. La forma en que funciona la memoria será objeto de un estudio en profundidad en un próximo artículo.

SPRITES EN MEMORIA

La definición de un *sprite* requiere 63 bytes de memoria para los 63 valores de las DATA. Sin embargo es conveniente disponer 64 bytes para cada *sprite*, ya que así se simplifican algo los cálculos. Estos 64 bytes se pueden almacenar en cualquier zona de la memoria disponible, siempre que sea un múltiplo de 64.

Se puede acceder a cada una de estas definiciones después de que ha sido almacenada, y para ello se utilizan los punteros de *sprite*. Hay ocho de estos punteros y cada uno de ellos se puede POKEar con un valor entre 0 y 255. Para identificar la posición del *sprite* se multiplica este valor por 64 (por esta razón la dirección de memoria tiene que ser un múltiplo de 64).

El valor máximo —255— representa pues el valor tope de 255 × 64, que es 16K, cubriendo por completo un bloque de memoria que es a lo que puede acceder el circuito integrado de video. (Existen cuatro de estos bloques o bancos de 16K, aunque sólo se puede acceder a ellos de uno en uno).

Los punteros son una poderosa herramienta que te permite acceder rápidamente a cualquier definición de *sprite* que tengas en memoria. De esta manera puedes llamar a toda una secuencia de *sprites*. Cada *sprite* reemplazará en la pantalla a su antecesor permitiendo obtener imaginativos efectos de animación.

Esta es de hecho una forma mucho más eficiente de utilizar los *sprites*: conmutando los punteros en vez de conmutar los sprites consigues que éstos estén disponibles para otros usos.

Los punteros de *sprites* siempre están situados en el extremo no utilizado de la memoria de pantalla de 1K, que normalmente corresponde a las direcciones 2040 a 2047. Puedes ver un ejemplo de esto en la línea 30 del programa del pájaro volador, en que se utiliza 2040.

Es posible reubicar la memoria de pantalla y mover también de esta forma los punteros de *sprite*. Los nuevos valores de los punteros tendrán que ser en este caso POKEados en las nuevas direcciones de los punteros.

DONDE TRANSCURRE LA ACCION

Antes de que puedas controlar el *sprite*, tendrás que mantener una lu-

ALMACENAMIENTO EN CINTA

Uno de los problemas menos conocidos que se presentan al trabajar con *sprites* es el de intentar almacenar los programas en cinta. Esto sólo ocurrirá si has cometido el comprensible error de ejecutar, durante el desarrollo del mismo, un programa que contenga un *sprite*. Podrías pensar que se trata de algo bastante normal, pero es una de las fuentes de los desconcertantes errores que se presentan al utilizar el SAVE y que con mucha ligereza se achacan a la unidad de cintas **C2N**.

La única solución segura al problema es hacer que tu comando normal de SAVE vaya precedido del siguiente POKE:

POKE 53269,0:SAVE «PROGNOMBRE»

Con esto se inhibe la presentación del *sprite* y se elimina la fuente de interferencias que afecta a la rutina de almacenamiento en cinta. Como alternativa, también puedes añadir el POKE al final del propio programa.

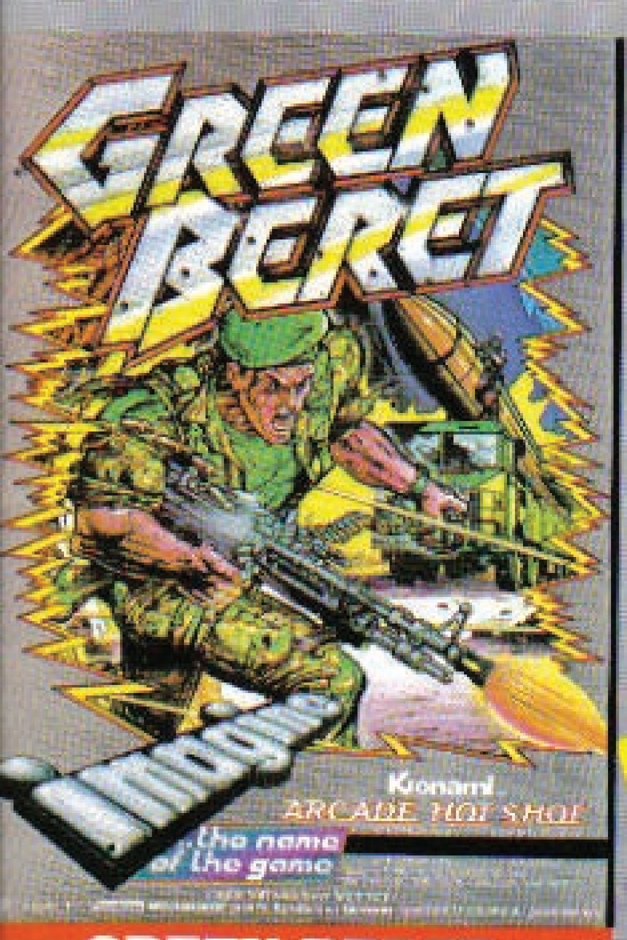
Tienes que utilizar la misma instrucción de POKE antes de la instrucción LOAD, cuando cargues un nuevo programa en tu **Commodore** después de la visualización de un *sprite*. No hace falta que hagas esto si efectúas un *reset* frío del sistema, lo que normalmente se hace apagando la máquina y volviendo a encenderla.

Los usuarios de disco estaréis muy felices de saber que los problemas relacionados con las instrucciones SAVE y LOAD y los *sprites* no se presentan en estas unidades, y podéis usar tranquilamente los procedimientos normales.



"V"

La serie de ciencia-ficción televisiva invade tu ordenador. La Tierra es visitada por alienígenas de aspecto humano. Sin embargo, estos reptiles pronto revelarán su objetivo final... conquistar la Tierra y esclavizar a la raza humana.



GREEN BERET

Green Beret (boina verde), es el programa de mayor acción que hayas visto jamás en un ordenador.
Decir Green Beret es decir un hombre perfectamente entrenado para la lucha. Prepárate...

THE WAY OF THE TIGER

Entra en el mundo de los samurais. Mantén la calma mientras el movimiento y las rutinas de combate te transportan a niveles que nunca pensaste posibles. Experimenta los sorprendentes efectos del "Triple Scroll" mientras intentas mejorar tus técnicas de lucha cuerpo a cuerpo, con espada samurai o con mil posibilidades más.

cha a brazo partido con el funcionamiento del **VIC-II 6566**, que es el *chip* de video que utiliza el **Commodore 64**. En particular tienes que saber cómo acceder a los comandos de control.

Esto no es tan terrible como pudiera parecer y ya nos ocuparemos de ello con mayor detalle en un próximo artículo. Por el momento echa una ojeada a la tabla 2, que lista las 47 direcciones de memoria desde 53248 a 53294, que son las utilizadas en la programación de los sprites. La expresión utilizada de V + valor pone de manifiesto la relación existente entre todas las direcciones que intervienen, tomando como base la dirección de partida, donde V = 53248.

A primera vista la tabla puede parecer escalofriante; sin embargo es una referencia esencial si quieres sacar el máximo partido a la programación de *sprites*. La notación de V + valor es más fácil de recordar que las direcciones específicas, y resulta más eficiente en la utilización de memoria, lo que será especialmente útil en los programas largos. Además su uso simplifica considerablemente la identificación de las instrucciones específicas de cada *sprite*, lo cual es una ayuda útil cuando se programan cambios y se hace necesaria una depuración.

Como puedes ver, cada uno de los ocho *sprites* normales van numerados y el valor de este número tiene importantes consecuencias.

En la tabla también puedes ver que hay direcciones para controlar todo lo referente al *sprite*, desde su posición hasta la detección de colisiones y el color.

El verdadero uso de la notación V + valor se describe con mucho más detalle en la próxima parte de este artículo, en la que nos ocuparemos de la forma interna de trabajar del control de *sprite*. No obstante si sientes curiosidad, echa una mirada al uso de los valores V + valor en el programa del pájaro que vuela.

En la línea 25 aparece V + 21, que es el comando de activación o habilitación del *sprite*. Los valores V de la línea 70 definen las coordenadas X e Y del sprite en la pantalla. En la línea 2400 y otras más del programa de la estación espacial tienes otros ejemplos que puedes analizar, refiriéndote a la tabla anterior.

Añade estas líneas extras de programa al programa del texto para tener un sencillo juego de pájaros y tiros con ayuda de tu recién definido *sprite*.

```
15 POKE 53280,2:POKE 53281,2
   :POKE 650,128:FOR Z=16000
   TO 16000+64*2:POKE Z,0
16 NEXT
20 FOR I=16000 TO 16077:READ
   A:POKE I,A:NEXT I:TI$=
   "000000":POKE V+29,1
25 POKE 2040,250:POKE 2041,
   251:POKE V+21,3:XX=100:
   YY=100:POKE V+40,1
30 FOR Z=1 TO 5:PRINT"[CLR/
   HOME][CTRL+5]TIEMPO
   [CTRL+1]"VAL(TI$):IF RND
   (1)>.3 THEN POKE V+23,RND
   (1)*2
35 A=INT(RND(1)*3)+1:X=X+10:
   IF X>239 THEN X=30
40 IF A=1 AND Y>70 THEN Y=Y-
   10
45 IF A=2 AND Y<200 THEN Y=Y
   +10
50 POKE V,X:POKE V+1,Y:POKE
   V+39,7:GET Z$:IF Z$="Z"
   AND XX>30 THEN XX=XX-5
55 IF Z$="X" AND XX<250 THEN
   XX=XX+5
60 IF Z$="P" AND YY>50 THEN
   YY=YY-5
65 IF Z$="L" AND YY<220 THEN
   YY=YY+5
70 POKE V+2,XX:POKE V+3,YY:IF
   PEEK(V+30)=3 THEN S=S+1
75 PRINT"PUNTOS:";S:NEXT Z:IF
   VAL(TI$)<59 THEN 30
80 PRINT"[CLR/HOME][CTRL+2]
   [CTRL+5]VECES[CTRL+5]!";
   END
115 DATA 8,0,0,8,0,0,62,0,0,
    8,0,0,8,0,0
```

GANADORES DE LOS MEJORES DE INPUT COMMODORE

En el sorteo correspondiente al número 8 realizado entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
Nicolás Sáez Ruiz	Benidorm (Alicante)	Dessert Fox
José Burrezo Alvarez	Málaga	Karateka
Antonio Palacios Cibrián	Madrid	R. of Impossibility
Josep Corominas Trujols	Palau Plegamans (Barcelona)	Commando
Marcelino Santos Aguadero	Santurce (Vizcaya)	Gi-Joe
M. Angel Torres Villaseca	El Palmar (Murcia)	Soccer
Juan Ruiz Alabarce	Mataró (Barcelona)	Green Beret
Carlos Gayubo Moreo	Algeciras (Cádiz)	Rambo
M.ª Nieves Somolinos López	Barcelona	Ley del Oeste
Jorge Manuel Pérez Vázquez	Cerdanyola Vallés (Barcelona)	Super Test

COMMODORE, COMPAÑIA AMERICANA, LIDER MUNDIAL EN NUMERO DE ORDENADORES INSTALADOS.





Más 64's que nadie.

El C-64 de Commodore sigue siendo el ordenador personal más vendido del mundo por sus prestaciones y posibilidades.

Más periféricos que nadie La Gama de periféricos y accesorios del C-64 multiplica sus funciones de una forma casi ilimitada: impresoras, unidades de disco, monitores... Todo un mundo informático a su alrededor para que usted le saque todo el partido.

Más software que nadie El C-64, por ser el ordenador más vendido, ha hecho que las compañías de software se vuelquen en él, creando un parque de programas que hoy le convierten en el 64 con más software del mercado. Software que abarca todos los sectores, desde los negocios hasta el educativo.

Le podemos asegurar que hoy por hoy el programa que usted necesita ya lo tiene el Commodore 64.

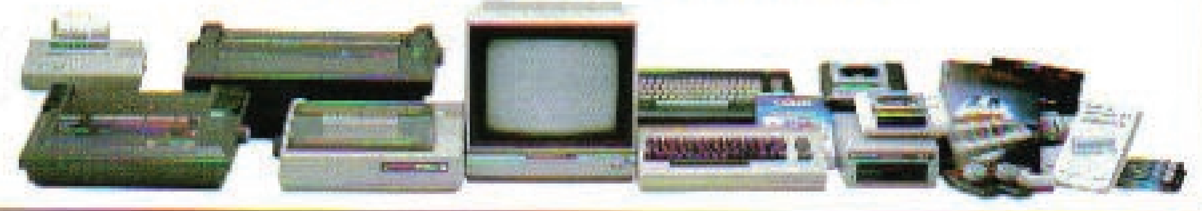
Más información que nadie El C-64, lejos de quedarse atrás y porque cada vez son más los que lo eligen, dispone del mayor número de publicaciones exclusivas, así como libros de documentación en castellano donde se tratan temas de interés, nuevos programas, nuevas ideas, nuevas aplicaciones...

Cada vez más gente investiga y se preocupa por su Commodore 64.

Y menos precio que nadie Y todo esto a un precio realmente sorprendente y asequible.

Acérquese hoy mismo a un distribuidor Commodore y entérese de su precio. Descubrirá qué fácil es poseer el mejor ordenador personal del mundo. El más vendido.





Microelectrónica y Control, s. a. *c/ Valencia, 49-53 08015 Barcelona - c/ Ardemans, 8 28028 Madrid*
Unico representante de Commodore en España

PUERTA ABIERTA A LA MICRORROBOTICA

La robótica ya no es un campo cerrado únicamente a las inversiones millonarias de los laboratorios de experimentación. Por el contrario, sistemas del tipo que vamos a describir resultan ser una inmejorable herramienta de alto componente educativo.

Todo el mundo permanecía callado. Un expectante silencio se adueñó de la sala. Sólo faltaba el retoque final a la que debía ser la última línea de un sencillo programa de demostración en BASIC.

R...U...N, RETURN y los pequeños motores eléctricos comenzaron a emitir su zumbido característico. Con sorprendente precisión, aquel brazo se dirigió sin vacilar hacia la caja de cerillas que reposaba de pie sobre el banco de trabajo. La pinza se cerró como si de los dedos pulgar e índice se tratase. En una nueva combinación de ágiles movimientos, el brazo electromecánico adoptó la postura del deportista victorioso, mostrando la caja que sostenía en su extremo, queriendo parecer una expresión del orgullo de sus multiples posibilidades.

A pesar del éxito de la prueba, todos sabemos que un robot no puede tener sensaciones humanas como las que pretendíamos adivinar. En realidad quien estaba totalmente ufano era el programador que momentos antes tecleaba ansiosamente frente a su **Commodore 64**.

La industria microrrobótica está a un nivel equivalente al que alcanzó la microinformática hace una decena de años, cuando el microordenador empezaba a popularizarse a precios asequibles (al menos en los paises anglosajones). Igual que ocurrió con los micros, es muy probable que quienes hoy disponen de uno no tardarán en comprar un robot doméstico, aunque son mercados distintos que no tienen por qué llevar un desarrollo parejo.

Los robots, lejos de representar un peligro para la futura mano de obra, están siendo orientados hacia la mejora de la calidad de vida humana por la vía de las herramientas de gran nivel de sofisticación. Se estima que cerca del sesenta por ciento de los robots instalados en plantas industriales realizan tareas netamente rutinarias o nocivas para un operador humano.

El **Brazo Robótico** fabricado por la firma británica **Memoco Electron** es una réplica —en material plástico— bastante afortunada de un modelo de grandes dimensiones, empleado en las cadenas de montaje de algunas fábri-





Commodore 128

Más 128 que nadie.

Commodore presenta el 128 más completo del mercado. El Commodore 128.

Un ordenador nacido para convertirse en mito.

Más prestaciones que nadie Para ser más que nadie hay que demostrar la capacidad de actuación. Para el C-128 estos son sus poderes:
• 122.365 Bytes libres en modo Basic • Biblioteca de programas más extensa del mercado (pues dispone de todos los programas del C-64, del C-128 y de CP/M® 3.0). • Teclado numérico independiente • Alta resolución • 80 columnas en pantalla • Compatible con la periferia del C-64.

En una palabra, el ordenador más completo de la gama 128.

Más ordenador que nadie Además y por si fuera poco, el C-128 es el único ordenador capaz de actuar como tres.

Primero como un C-64, con cuyos programas y periféricos es compatible, segundo como un 128 en toda la extensión de la palabra; y tercero, como un ordenador con sistema operativo CP/M®.

Y todo, con sólo pulsar una tecla.

Ha comenzado la era de los 128, conózcalos y sepa que uno ya es más 128 que nadie, el C-128.

PRINCIPALES CARACTERISTICAS
Microprocesadores: 8502 (1 ó 2 MH$_z$); Z80A (4 MH$_z$); MMU para gestión de memoria. – RAM total de 128 Kb. – 122 Kb de RAM libres en modo BASIC. – ROM 48 Kb + 20 Kb. – Pantalla texto de 80 × 25 y 40 × 25. – Máxima resolución 640 × 200. – 16 colores y los 16 a la vez en pantalla. – 3 voces con control de envolvente y 8 octavas. – Teclado de 92 teclas con módulo numérico independiente.







Microelectrónica y Control *c/ Valencia, 49-53 08015 Barcelona - c/ Ardemans, 8 28028 Madrid*
Unico representante de Commodore en España.

® es una marca registrada de Digital Research

cas. Pero no es sólo una maqueta, sino un mecanismo completamente operativo.

GRADOS DE LIBERTAD

Un grado de libertad es, en términos simplistas, un tipo de movimiento elemental que puede realizar un sistema mecánico, tal como un giro o un desplazamiento en ambos sentidos. Cuanto mayor es el número de ellos, mayor es la sofisticación de un robot.

El **Brazo Robótico** dispone de seis grados de libertad. Esto significa que puede acometer seis tipos de movimiento elemental y que el desplazamiento resultante es igual a la combinación de todos y cada uno de ellos. Si comparamos este sistema con un brazo real nos encontramos con que puede abrir y cerrar la pinza (1er. grado) como si se tratase de los dedos pulgar e índice. Puede hacerla girar en torno a un eje (2º grado), igual que la muñeca. La muñeca puede asimismo efectuar un giro (3er. grado) en dirección perpendicular a la anterior. El antebrazo se desplaza (4º grado) en base al giro del codo. El brazo une al codo con el hombro, desde donde puede elevarse o descender a todo el conjunto (5º grado) y el último movimiento consiste en un giro cuyo eje es perpendicular a la base de apoyo y sustentación (6º grado).

CONSTRUCCION

A diferencia de algunos robots similares, que utilizan profusamente cuerdecillas, muelles y sistemas de polea, el **Brazo Robótico** utiliza exclusivamente engranajes, confiriéndole una mayor robustez y fiabilidad al conjunto.

Todo el brazo se apoya sobre una base/carcasa de plástico azul grisáceo, en cuya parte frontal aparece el conector que sirve para unirle con el controlador y, justo a su lado, encontramos un *jack* destinado al cable que procede del transformador de alimentación eléctrica.

En la base hay cuatro separadores de goma, para facilitar la adherencia a la mesa de trabajo, y asoma un potenciómetro de control, de igual aspecto que los otros cuatro distribuidos por el brazo. Igualmente, aparece una tapa que oculta un compartimento para un par de pilas de voltio y medio.

La pinza, de color naranja, es accionada por un motor eléctrico, lo cual permite diferentes grados de apertura. Dos láminas de caucho rugoso le proporcionan mayor nivel de adherencia con las piezas que atrapa. Una de ellas oculta un microrruptor que detecta la llegada a un determinado umbral de presión sobre el objeto asido.

La muñeca es similar a un grueso disco de plástico azul que incluye el soporte de las piezas que forman la pinza. Este mismo conjunto es utilizado por un simpático y simple robot de control manual, muy popular entre la chiquillería británica.

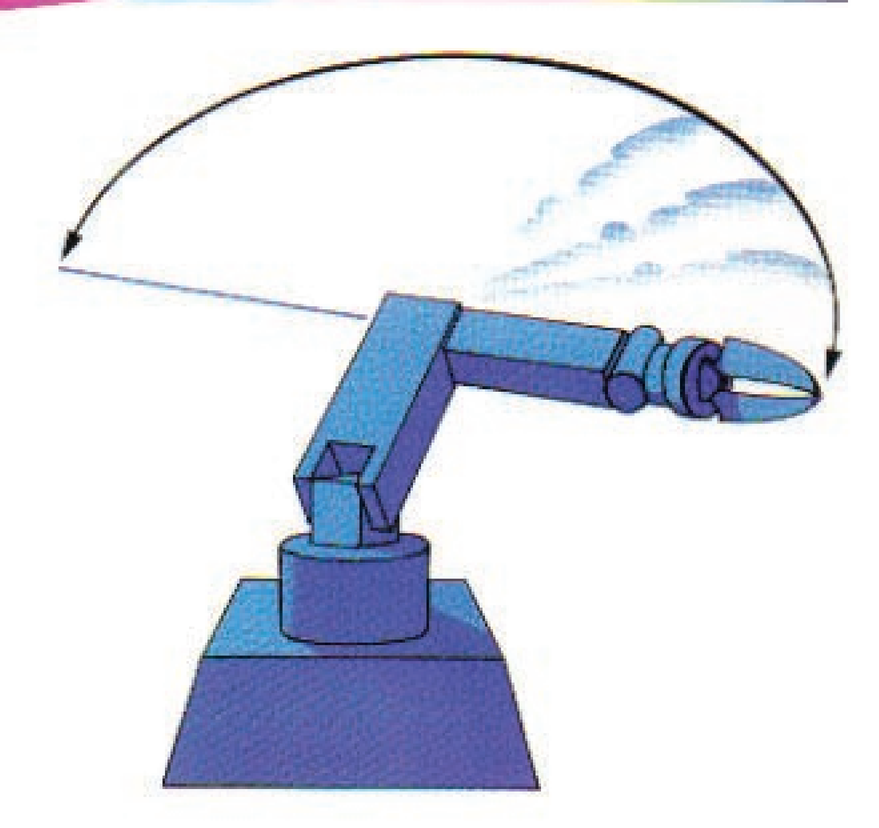
El resto del brazo son piezas ensambladas de plástico gris. A través de las zonas de unión que tapan a los ejes asoman tímidamente algunos engranajes.

MOTORES

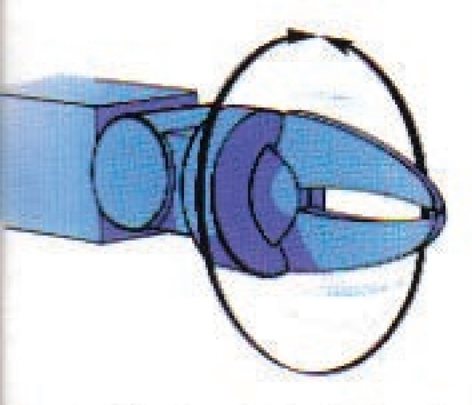
El costo del sistema que genera el movimiento es un factor crítico a la hora del diseño de un sistema robótico. Por otro lado, los sistemas oleohidraúlicos (utilizan aceite) y neumáticos (el aire comprimido es el elemento básico) son los que proporcionan mayor precisión de movimientos.

También sucede que estas tecnologías son más caras y difíciles de aplicar que los motores eléctricos.

Los motores eléctricos empleados en esta categoría de sistemas robóticos se encuadran fundamentalmente en dos categorías. El primero está formado por los motores paso-a-paso, capaces de producir un giro completo en un determinado número de pasos (2, 4, 6... etc.) controlables por circuitería electrónica. Así, un motor de 24 pasos gira 15 grados por vez. Aplicado este movimiento a un sistema de engranajes reductores, se produce un movimiento extremadamente preciso, con la peculiaridad de que es más fácil controlar el resultado final. Pero no todo son ventajas, el sistema mecánico acoplado a él puede «patinar» en algún momento y echar por tierra nuestro maravilloso sistema de control. Un inconveniente es el elevado precio de estos motores.



El otro grupo son los motores convencionales, que a igualdad de tamaño suelen proporcionar más rendimiento que los paso a paso y también son más baratos. La otra cara de la moneda es la dificultad de conocer la



magnitud del giro efectuado. Para ello hay que prever algún sistema de control, del que hablaremos cuando discutamos el concepto realimentación. Pues bien, el **Brazo Robótico** emplea esta última categoría de motores, que parece obligado pensar que se debe a obtener un buen precio de salida.

RESUMEN DE MOVIMIENTOS

Pinzas
* Abiertas o cerradas, con todas las posiciones intermedias.

Muñeca
* Giro entre 0 y 270 grados.
* Subida y bajada en aprox. 145 grados.

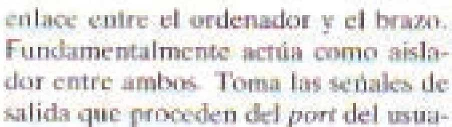
Antebrazo
* Giro en 180 grados.

Brazo
* Subida y bajada en aprox. 65 grados.

Hombro
* Giro en aprox. 270 grados.

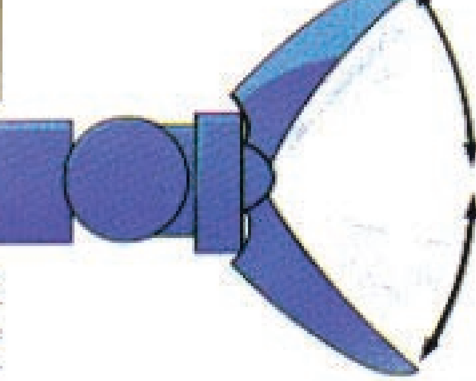
EL CONTROLADOR

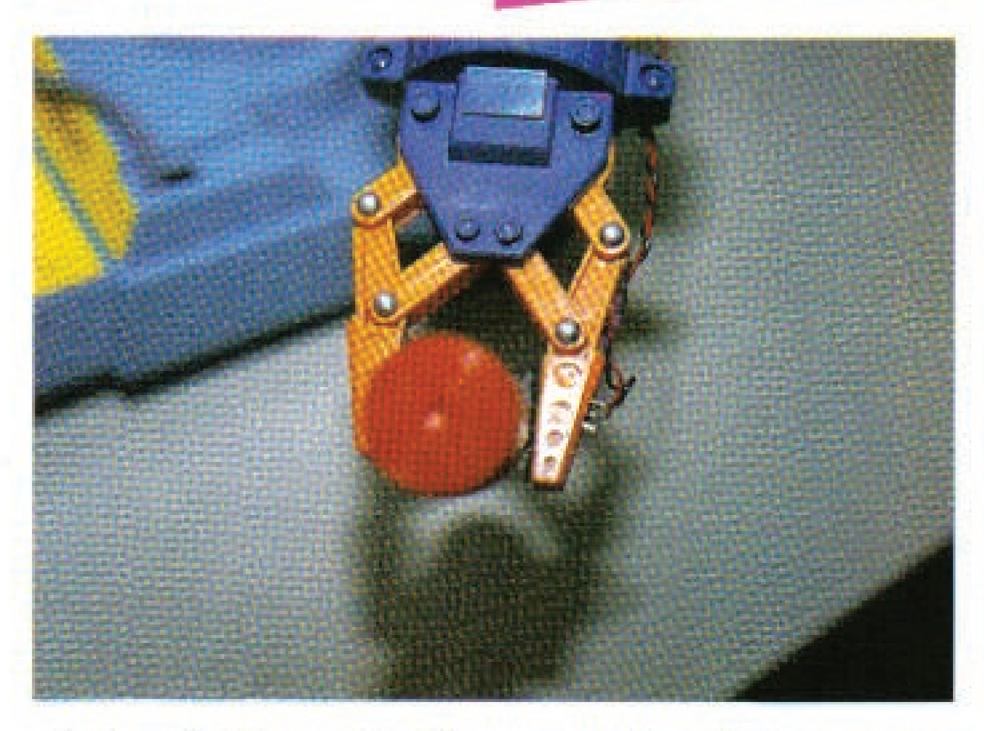
El controlador es un circuito electrónico que actúa como elemento de



enlace entre el ordenador y el brazo. Fundamentalmente actúa como aislador entre ambos. Toma las señales de salida que proceden del *port* del usuario del **Commodore 64** y activa una serie de relés que al cerrar su circuito, ponen en marcha su motor correspondiente. Su corriente de alimentación la recibe desde el propio ordenador.

Por un extremo de la tarjeta de circuito impreso surge un conector, al que llegan los cables planos que provienen del mencionado *port* y del brazo respectivamente. Por otro lado un cable negro va al *port* 1, destinado a *joysticks* y *paddles*, para que el ordenador pueda transformar el valor analógico de la resistencia de los potenciómetros de realimentación en cantidades digitales manejables por el ordenador.





En lugar de relés se podrían haber utilizado transistores de potencia, por ejemplo, obteniéndose un controlador de «estado sólido» (sin piezas móviles), pero no es recomendable, porque al no separarse físicamente los circuitos del ordenador y los motores, podrían surgir averías en el ordenador. Aunque un controlador que utilizase optoacopladores obviaría ese problema y produciría menos ruido...

¿QUE ES LA REALIMENTACION?

Realimentación es un antiguo concepto muy utilizado en electrónica y estadística. Quiere decir que un determinado sistema recoge parte de lo que produce en su salida y lo aplica como parte de su entrada, alimentándose con ello. Un ejemplo concreto somos nosotros mismos, que la empleamos sin saberlo. Supongamos que deseamos tomar un vaso que está encima de la mesa. La entrada de datos es el ojo, que nos indica la posición del mismo y la distancia que lo separa de nuestra mano, siendo la salida que proporciona el cerebro una serie de estímulos que hacen moverse al brazo y la mano. A medida que aproximamos la mano al objetivo, el ojo indica al cerebro que estamos consiguiéndolo y por tanto la cantidad y la calidad del movimiento que realiza el brazo. Una vez alcanzado, el sistema ojo-cerebro-brazo no necesita hacer nada más. A este tipo de realimentación se le llama negativa, pues a mayor alejamiento del objetivo es necesario más movimiento.

Por el contrario, si hubiera un foco de calor, una cerilla por ejemplo, de lo que se trataría sería de alejar la mano lo más posible del foco de calor. Esto se llama realimentación positiva, que supone que cuanto más cerca estamos, más movimiento se necesita.

Antes decíamos que el **Brazo Robótico** incorpora cinco potenciómetros y un microrruptor. Pues bien cada potenciómetro es una especie de ojo, capaz de informar al **Commodore 64** de la posición en la que se halla cada uno de los elementos (asociarlo con los grados de libertad) del brazo. El cursor está solidariamente unido al eje de giro de cada uno de los engranajes que producen el movimiento. Como el potenciómetro no es más que una resistencia variable en función de la posición de su cursor, basta con conocer la resistencia eléctrica que presenta un potenciómetro para saber la posición de un elemento del brazo. Conociendo las resistencias de los cinco potenciómetros en las distintas posiciones deseadas, no hay más que poner a trabajar a los motores hasta que la lectura de los potenciómetros arroje el resultado que buscamos.

La pinza trabaja de un modo similar, sólamente se diferencia en que la resistencia que presenta el interruptor es cero o infinita, por lo que el motor girará hasta que se detecte la activación del mismo.

En teoría esto es sencillo, pero hay que ser minucioso por la complicación que supone trabajar simultáneamente hasta con cinco parámetros. También es deseable que el desplazamiento se realice gradualmente, en lugar de a saltos.

FUNCIONAMIENTO

Una vez hechas las conexiones entre el ordenador y el brazo, estamos listos para comenzar a trabajar, porque no es necesrio ajuste previo alguno.

El *port* del usuario del **Commodore 64** puede ser programado para que sus líneas actúen como entrada o salida. En este caso, el *port* B utilizará sus ocho líneas como salida, para controlar el movimiento de cada motor. Una vez programado el DDRB (dirección 56579), simplemente con distintos POKEs a la dirección 56577 se hace girar un motor (o varios al mismo tiempo), e incluso controlar la velocidad de cada uno de ellos, también realiza una tarea complementaria para la lectura de los potenciómetros, que será leída cuando nos dirijamos a un determinado motor previamente desactivado mediante el bit 2 del *port*.

El manual de instrucciones es breve y sucinto, pero incluye toda la información útil para sacarle el mayor partido al brazo. A modo de ejemplo incorpora dos interesantes programas de aplicación, en los que se incluye una interesante rutina en código máquina para la lectura de un *joystick*, que permite instruir al robot en los movimientos que deberá efectuar posteriormente.

Desde luego, el **Brazo Robótico** demostró ser mucho más de lo que en principio parecía un juguete caro.

SPRITES DE COLOR EN EL COMMODORE

- DEFINICION DE LOS SPRITES MULTICOLOR
- USO DEL GENERADOR DE SPRITES
- REGISTROS DE CONTROL

Con este programa generador de *sprites* multiuso podrás crear ficheros especiales de datos para usar en tus propios programas. Aprovecha la ocasión para echar una ojeada a los importantes registros del VIC.

Un *sprite* es un tipo de gráfico definido por el usuario extremadamente versátil que confiere al **Commodore 64** un potencial tremendo en la programación de juegos de todo tipo.

Aunque sencillos en teoría, el aprovechamiento a fondo de esta poderosa herramienta requiere un conocimiento profundo de la forma de funcionar del *chip* **VIC-II**, en especial de las aproximadamente treinta posiciones de memoria relacionadas con la forma física, color y movimiento de los ocho *sprites* que trae como número estándar.

Muchos de los aspectos generales ya han sido discutidos anteriormente así como la forma detallada de definir *sprites* de un solo color. En este artículo veremos la manera de crear *sprites* multicolor, y cómo poner en acción las definiciones de *sprites*.

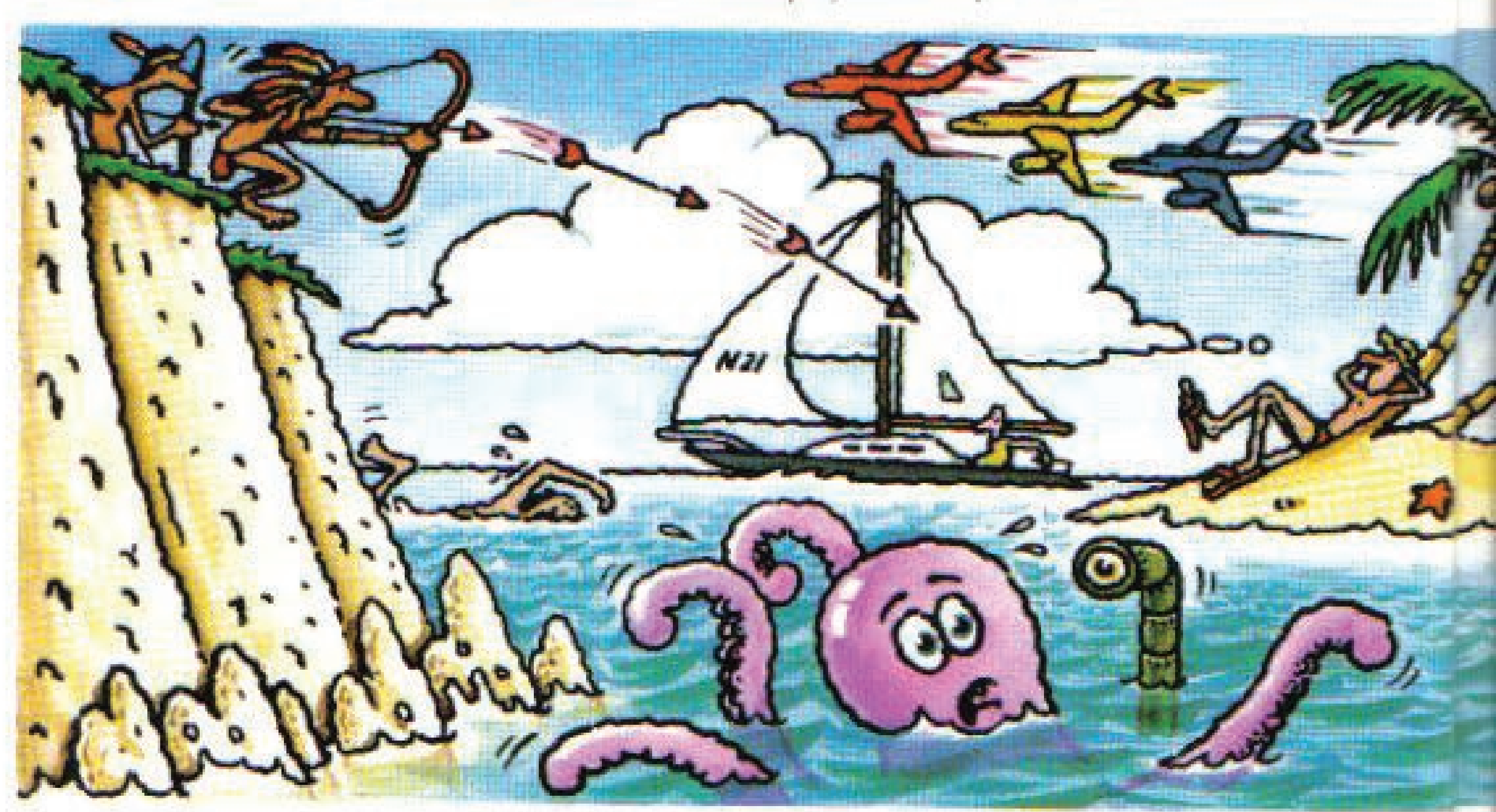
También veremos la manera de crear unos cuantos sprites para representar barcos, animales marinos, una isla desierta y otros «personajes» que serán utilizados en un artículo posterior como base para una animada puesta en escena; precisamente el tipo de cosas en que están basados muchos juegos comerciales.

SPRITES MULTICOLOR

Los *sprites* multicolor pueden definirse de una forma análoga a los *sprites* de un solo color de que ya hemos hablado. Y, al igual que antes, el programa de utilidad que sigue más adelante te permitirá en breve que los configures fácilmente sobre la pantalla.

Los *sprites* multicolor pueden contener hasta tres colores. Como nada se obtiene a cambio de nada, en el proceso habrás de sacrificar algo la resolución horizontal, ya que cada *pixel* tendrá una anchura doble de la normal y se necesitan dos bits del *sprite* para definirla. Hacen falta doce pares de bits para definir cada una de las 21 líneas horizontales de un *sprite* multicolor.

Cada uno de estos pares puede tomar una de cuatro configuraciones: 00, 01, 10 o 11. Cada una de ellas se utiliza para dar una información específica acerca del doble *pixel* al que representa, como se ve en la tabla 1.



```
00-El pixel de doble
   anchura adopta el color
   del fondo,
   quedando invisible.
01-Establece el color
   especificado en el
   registro V+37.
10-Establece el color
   "unico" para el sprite
   multicolor, visuali-
   zando el color
   especificado en los
   registros V+39 a V+46.
11-Establece el color
   especificado en el
   registro V+38.
```

GENERADOR DE SPRITES

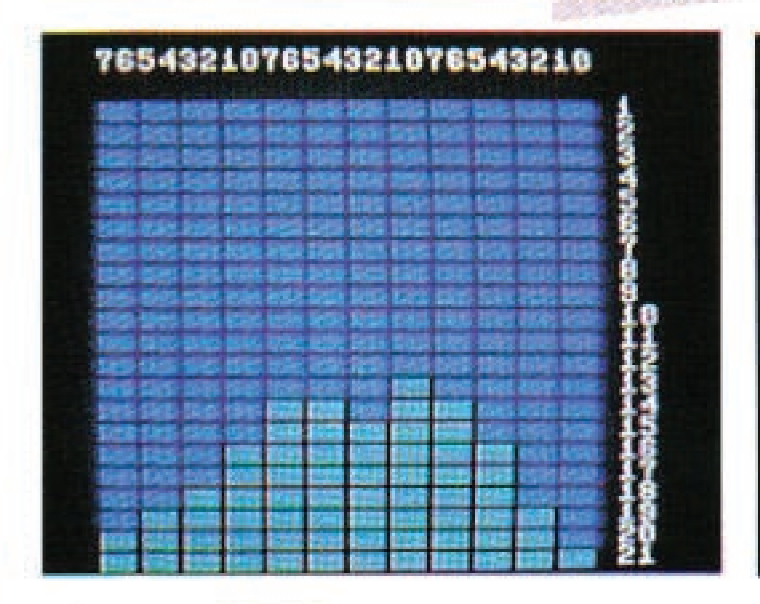
El siguiente programa se puede utilizar para definir *sprites* multicolores o normales de alta resolución, almacenándolos en ficheros especiales para datos de *sprites*. Por cada fichero se pueden definir hasta 64 *sprites*. Cuando el fichero está parcial o totalmente lleno, puedes guardarlo en cinta o en disco. Puedes crear cualquier número de ficheros de datos separados pero asegúrate de que le das a cada uno un nombre distinto.

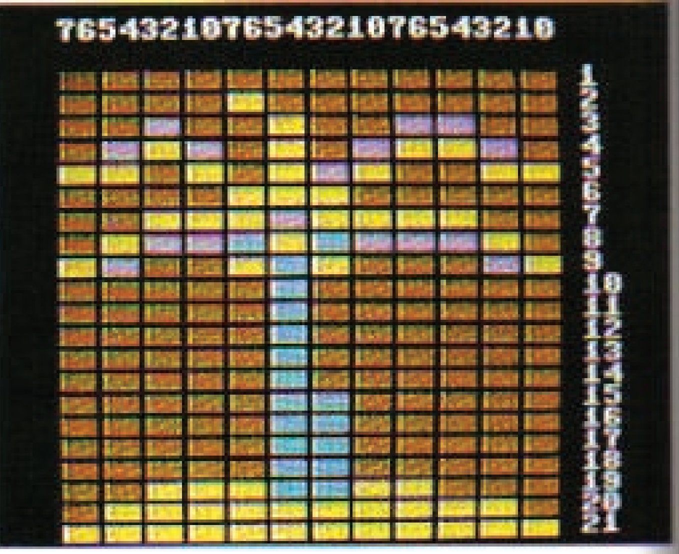
Este programa crea un área de almacenamiento para *sprites* donde se lee, y a partir del cual se escribe durante la creación de cada *sprite*. Tienes la opción de examinar cualquiera de los 64 diseños de *sprites*, editarlo, o volcar los valores deseados en forma de DATAs en pantalla o en una impresora, para después copiarlos e incluirlos en tus propios programas.

Los ficheros de datos de los *sprites* se almacenan en forma de bytes entre las direcciones de memoria 12288 y 16384 y puedes llamarlos directamente desde tus propios programas.

Observa que alguna de las últimas líneas contienen símbolos que son la segunda letra de la abreviatura de POKE, y parecen algo así como pO cuando la pantalla se ha cambiado al modo de minúsculas (pulsa SHIFT y COMM al mismo tiempo). Hay que utilizar estas abreviaturas para que la longitud de las líneas se mantenga dentro del límite de 80 caracteres. Date cuenta también de que los símbolos que aparecen en las líneas 180 y 190 se obtienen pulsando SHIFT más



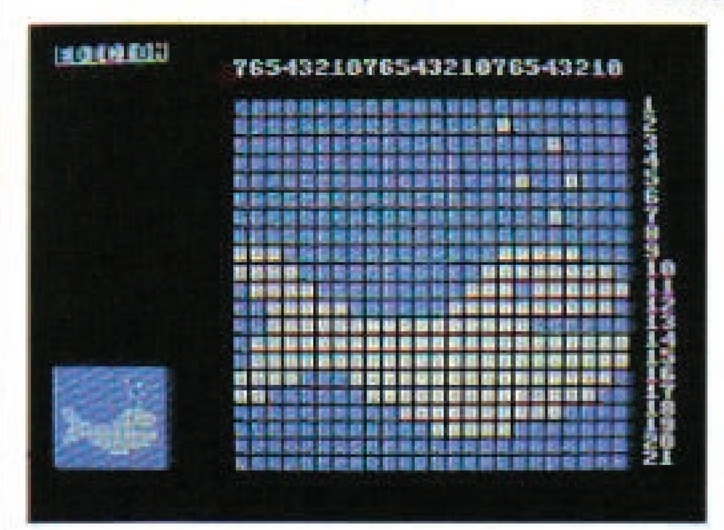


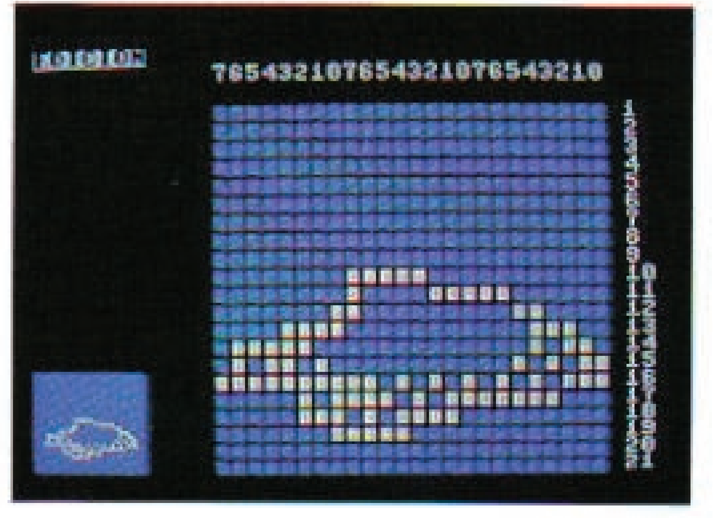


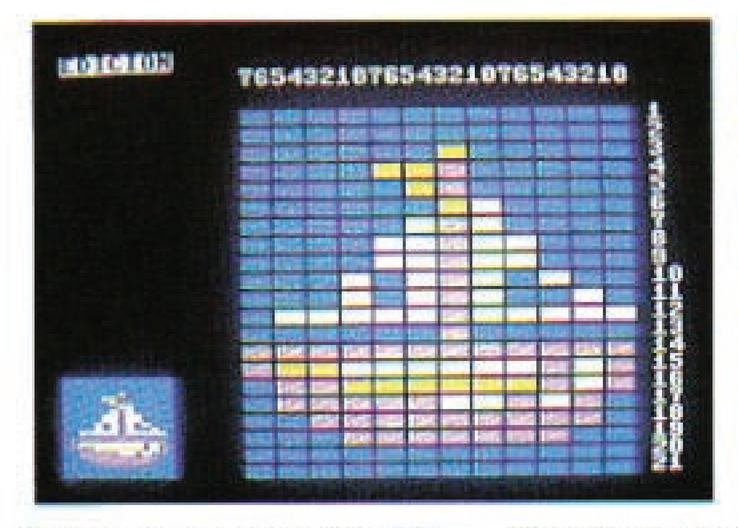
[O], junto con [COMM] más Y en la segunda línea. Estas líneas definen la cuadrícula de caracteres del *sprite*. Los restantes símbolos están relacionados con el «modo de cita», los cambios de color y los controles del cursor.

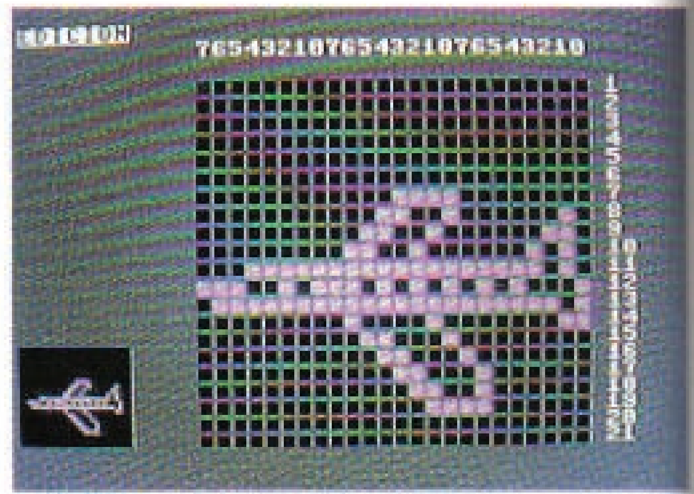
```
10 POKE 51,255:POKE 52,47:
   POKE 55,255:POKE 56,47:
   CLR
20 POKE 53280,6:POKE 53281,6
   :PRINT "[SHIFT+CLR/HOME]
   [CTRL+4]"TAB(12);
   "[13*COMM.+P]";CHR$(8)
30 PRINT TAB(12);"[CTRL+9]
   EDITOR DE SPRITES
   [CRSR abajo]"
40 CH=1:PRINT "[CLR/HOME]
   [3*CRSR abajo][COMM.+4]
   QUIERES SPRITES MULTI
   COLOR":INPUT"(S/N)";A$
50 IF A$<>"S" AND A$<>"N"
   THEN 40
60 IF A$="S" THEN CH=2:
   GOTO 90
70 INPUT"[CRSR abajo]ESCRIBE
   COLORES DEL SPRITE";CL(1)
   :IF CL(1)<0 OR CL(1)>15
   THEN 70
80 CL(2)=CL(1):CL(3)=CL(1):
   GOTO 130
90 FOR Z=1 TO 3
100 PRINT"[CRSR abajo]
    ESCRIBE COLOR DEL
    SPRITE";Z;
110 INPUT CL(Z):IF CL(Z)<0
    OR CL(Z)>15 THEN 100
120 NEXT Z
130 INPUT"][CRSR abajo]
    ESCRIBE COLOR DE FONDO";
    CL(4):POKE 650,128
140 IF CL(4)<0 OR CL(4)>15
    THEN 130
150 IF CL(4)=0 THEN POKE
    53280,11:POKE 53281,11:
    GOTO 170
160 POKE 53280,0:POKE 53281
    ,0
170 FOR Z=832 TO 894:POKE Z
    ,0:NEXT Z
180 L=1155:X=0:Y=0:CC=207:
    DD=24:G$="[SHIFT+O]":
    C(1)=207:C(2)=207:
    C=55427
190 IF CH=2 THEN G$="[SHIFT
    +O][COMM+Y]":DD=12:C(2)=
    247
200 A(1)=128:A(2)=64:A(3)=32
    :A(4)=16:A(5)=8:A(6)=4:
    A(7)=2:A(8)=1
210 V=53248:POKE 2040,13:
    POKE V+21,1:POKE V,28:
    POKE V+1,197:POKE V+28,0
220 POKE V+38,CL(1):POKE V+
    39,CL(2):POKE V+37,CL(3)
    :IFCH=2 THEN POKE V+28,1
230 POKE V+23,1:POKE V+29,1
240 PRINT"[SHIFT+CLR/HOME]
    [17*CRSR abajo]":POKE
    646,CL(4)
250 FOR Z=1 TO 6:PRINT
    "[CTRL+9][7*ESPACIO]":
    NEXT Z
260 FF$="":FOR Z=1 TO DD:
    FF$=FF$+G$:NEXT
270 PRINT"[CLR/HOME][CTRL+2]
    [11*CRSR izq.][CRSR
    abajo]7654321076543210 7
    543210[CRSR abajo]"
280 FOR Z=1 TO 21:PRINT
    "11*CRSR izq.][CTRL+9]";
    :POKE 646,CL(4)
290 PRINT FF$;"[CTRL+0]
    [CTRL+2]";Z:NEXT Z:POKE
    895,0
300 PRINT"[CLR/HOME][CTRL+9]
    EDICION"
310 GET A$:FOR Z=0 TO CH-1:
    POKE L+X+Y*40+Z,C(Z+1):
    NEXT Z
320 IF A$="E" THEN 1370
330 IF A$=CHR$(20) OR A$=" "
    THEN AA$=A$:A$="4":GOSUB
    580:A$=AA$
340 IF A$="R" THEN POKE V+21
    ,0:RUN
350 IF A$="*" THEN 1220
360 IF A$=CHR$(134) THEN
    POKE V+23,1
370 IF A$=CHR$(133)THEN POKE
    V+29,1
380 IF A$=CHR$(136)THEN POKE
    V+23,0
390 IF A$=CHR$(135)THEN POKE
    V+29,0
400 IF A$="[CRSR dcha.]"THEN
    X=X-CH
410 IF A$="[CRSR arriba]"
    THEN Y=Y-1
420 IF A$="[CRSR izq.]" OR
    A$=" " THEN X=X+CH
```

```
430 IF(A$="[CRSR abajo]" OR
    A$=CHR$(13)) THEN Y=Y+1
440 IF X<0 THEN X=24-CH
450 IF X>24-CH THEN X=0
460 IF Y<0 THEN Y=20
470 IF Y>20 THEN Y=0
480 IF A$=CHR$(13) THEN X=0
490 IF A$="S" THEN X=0:Y=0
500 IF A$="V" THENGOSUB 1080
510 CC=PEEK(L+X+Y*40)
520 IF A$="S" THEN GOSUB 770
    :GOSUB 690:GOSUB 770:
    GOTO 300
530 IF A$="C" THEN GOSUB 770
    :GOSUB 780:GOSUB 770:
    GOTO 300
540 FOR Z=0 TO CH-1:POKE L+X
    +Y*40+Z,32:NEXT Z
550 IF VAL(A$)>0 AND VAL(A$)
    <4 THEN GOSUB 580
560 IF A$="[SHIFT+CLR/HOME]"
    THEN FOR Z=832 TO 895:
    POKE Z,0:NEXT Z:GOTO 270
570 GOTO 310
580 XX=832+INT(X/8)+(Y*3):
    PP=PEEK(XX):PE=PEEK(XX)
    AND A(X-(INT(X/8)*8)+1)
590 PF=PEEK(XX) AND A(X-(INT
    (X/8)*8)+2)
600 IF CH=2 THEN 640
610 IF A$<>"4" AND PE=0 THEN
    POKE XX,PEEK(XX)+A(X-
    (INT(X/8)*8)+1)
620 IF A$="4" AND PE<>0 THEN
    POKE XX,PEEK(XX)-A(X-
    (INT(X/8)*8)+1)
630 GOTO 680
640 IF PE=0 AND(A$="1" OR A$
    ="2")THEN POKE XX,PEEK
    (XX)+A(X-(INT(X/8)*8)+1)
650 IF PE<>0 AND(A$="4" OR
    A$="3")THEN POKE XX,PEEK
    (XX)-A(X-(INT(X/8)*8)+1)
660 IF PF=0 AND(A$="1" OR A$
    ="3")THEN POKE XX,PEEK
    (XX)+A(X-(INT(X/8)*8)+2)
670 IF PF<>0 AND(A$="4" OR
    A$="2")THEN POKE XX,PEEK
    (XX)-A(X-(INT(X/8)*8)+2)
680 FOR Z=0 TO CH-1:POKE C+X
    +Y*40+Z,CL(VAL(A$)):NEXT
    Z:RETURN
690 GOSUB 770:NU=0:PRINT
    "[CLR/HOME][CTRL+9]
    SPRITE NO."
700 NU$="":INPUT"[CLR/HOME]
    [2*CRSR abajo]0-63";NU$:
    NU=VAL(NU$):IFNU<0ORNU>
    63OR LEN(NU$)=0 THEN 700
710 PRINT"[CRSR abajo][CTRL
    +9]PUNTERO:":PRINT192+NU
720 PRINT"[CRSR abajo][CTRL
    +9]DIR.INIC.:":PRINT
    12288+NU*64
730 PRINT"[CRSR abajo][CTRL
    +9]DIR.FINAL:":PRINT
    12288+NU*64+63
740 FOR Z=0 TO 63:POKE 12288
    +NU*64+Z,PEEK(832+Z):
    NEXT Z
750 PRINT"[CRSR abajo]>
    PRESIONA TECLA":PRINT
    "PARA CONTINUAR":POKE198
    ,0:WAIT198,1:POKE198,0
760 RETURN
770 PRINT"[CLR/HOME]";:FOR
    Z=1 TO 17:PRINT
    "[11*ESPACIO]":NEXT Z:
    RETURN
780 NU$="":PRINT"[CLR/HOME]
    [CTRL+9]SRSPRITE NO.":
    INPUT"[CLR/HOME]
    [2*CRSR abajo]0-63";NU$
790 NU=VAL(NU$):IF NU<0 OR
    NU>63 OR LEN(NU$)=0 THEN
    780
800 PRINT"[CLR/HOME][4*CRSR
    abajo]>COPIANDO"
810 NU=12288+NU*64:PO=832:IF
    CH=2 THEN 870
820 TT=0:FOR Z1=0 TO 20:FOR
    Z2=0 TO 2:FOR Z3=0 TO 7
830 XX=C+(Z1*40)+(Z2*8)+Z3
840 IF(PEEK(NU+TT)AND A(Z3+1
    ))<>0 THEN POKE XX,CL(2)
    :GOTO 860
850 POKE XX,CL(4)
860 NEXT Z3:POKE 832+TT,PEEK
    (NU+TT):TT=TT+1:NEXT Z2,
    Z1:RETURN
870 TT=0:FOR Z1=0 TO 20:FOR
    Z2=0 TO 2:FOR Z3=0 TO 7
    STEP 2
880 XX=C+(Z1*40)+(Z2*8)+Z3
890 R1=(PEEK(NU+TT)AND A(Z3+
    1))
900 R2=(PEEK(NU+TT)AND A(Z3+
    2))
910 IF R1<>0 AND R2<>0 THEN
    POKE XX,CL(1):POKE XX+1,
    CL(1)
920 IF R1<>0 AND R2=0 THEN
    POKE XX,CL(2):POKE XX+1,
    CL(2)
930 IF R1=0 AND R2<>0 THEN
    POKE XX,CL(3):POKE XX+1,
    CL(3)
```










```
940 IF R1=0 AND R2=0 THEN
    POKE XX,CL(4):POKE XX+1,
    CL(4)
950 NEXT Z3:POKE 832+TT,PEEK
    (NU+TT):TT=TT+1:NEXT Z2,
    Z1:RETURN
960 I$="":INPUT "[SHIFT+CLR/
    HOME](I)MPRESORA O (P)AN
    TALLA";I$
970 IF I$<>"I" AND I$<>"P"
    THEN 960
980 IF I$="I" THEN OPEN4,4:
    CMD4
990 PRINT"[CRSR abajo]DATAS
    DEL SPRITE";LU:LN=(192+
    LU)*64:PRINT
1000 FOR Z=LN TO LN+62:PRINT
     LEFT$(STR$(PEEK(Z))+
     "[3*ESPACIO]",4);:NEXTZ
1010 PRINT"[CRSR izq.]0":
     PRINT"[CRSR abajo]
     PUNTERO DEL SPRITE=";
     192+LU
1020 PRINT"[CRSR abajo]DIR.
     INICIAL =";LN
1030 PRINT"[CRSR abajo]DIR.
     FINAL =";LN+63
1040 IF I$="P" THEN PRINT
     "[CRSR abajo]>PRESIONA
     UNA TECLA PARA MODO
     EDICION"
1050 IF I$="P" THEN POKE 198
     ,0:WAIT198,1:POKE 198,0
1060 IF I$="I" THEN PRINT#4:
     CLOSE 4
1070 GOTO 170
1080 PRINT"[CLR/HOME][CTRL+9]
     EXAMEN":FOR Z=0 TO 63
1090 NU=12288+Z*64
1100 PRINT"[CLR/HOME]
     [16*CRSR abajo][CTRL+2]
     SPRITE[4*ESPACIO]
     [4*CRSR izq.]";Z
1110 POKE 2040,192+Z
1120 GET R$
1130 IF R$=CHR$(13)THEN 1210
1140 IF R$="B" THEN Z=Z-1:IF
     Z<>-1 THEN 1090
1150 IF Z=-1 THEN Z=63:GOTO
     1090
1160 IF R$="C" THEN NU=Z:
     GOSUB 800:GOTO 1210
1170 IF R$=" " THEN 1200
1180 IF R$="D" THEN POKE
     V+21,0:LU=Z:GOTO 960
1190 GOTO 1120
1200 NEXT Z:GOTO 1080
1210 POKE 2040,13:GOSUB 770:
     GOTO 300
1220 GOSUB 770:PRINT"[CLR/
     HOME][CRSR abajo]
     [3*CRSR izq.]QUIERES
     GUARDAR SPRITE (S/N)?"
1230 GET R$:IF R$="N" THEN
     1270
1240 IF R$="S" THEN 1260
1250 GOTO 1230
1260 PRINT"[CLR/HOME]
     [CRSR abajo]";:FOR Z=1
     TO 39:PRINT" ";:NEXT Z:
     GOSUB 690
1270 POKE V+21,0:POKE 53280,
     14:POKE 53281,6
1280 PRINT"[SHIFT+CLR/HOME]
     [COMM.+4]QUIERES
     GUARDAR(SAVE) O CARGAR
     (LOAD) LOS DATA (S/L)?"
     :S$=" LOAD"
1290 GET F$:IF F$="S" THEN
     S$=" SAVE":GOTO 1310
1300 IF F$<>"L" THEN 1290
1310 PRINT"[SHIFT+CLR/HOME]
     [CTRL+9]";TAB(13);
     "RUTINA";S$
1320 IFF$="L"THENPRINT
     "[CRSR abajo]LOAD "
     CHR$(34)"FICH. SPRITES
     1";CHR$(34);",1,1":
     GOTO1350
1330 PRINT"[CRSR abajo]P
     [SHIFT+0]43,0:P[SHIFT+0]
     44,48:P[SHIFT+0]45,0:
     P[SHIFT+0]46,64:";
1340 PRINT"[CRSR abajo]SAVE"
     ;CHR$(34)"FICHERO
     SPRITES 1";CHR$(34);",
     1,1"
1350 PRINT"[11*CRSR abajo]
     P[SHIFT+0]43,1:P
     [SHIFT+0]44,8:P[SHIFT+0]
     45,";PEEK(45);":P
     [SHIFT+0]46,";PEEK(46);
     ":RUN"
1360 END
1370 POKE V+21,0:PRINT
     "[SHIFT+CLR/HOME]
     [CTRL+9]ESCRIBE NUMERO
     DE SPRITE,SEGUIDO POR
     LOS DATOS"
1380 NU$="":INPUT"[CLR/HOME]
     [2*CRSR abajo]0-63";NU$
     :NU=VAL(NU$):IF NU<0OR
     NU>63ORLEN(NU$)=0THEN
     1370
1390 FOR Z=0 TO 63
1400 PRINT"[SHIFT+CLR/HOME]
     DATA";Z;:X=0;:INPUT X:
```

```
     IF X<0 OR X>255 THEN
     1400
1410 POKE(192+NU)*64+Z,X:
     NEXT Z
1420 PRINT"[SHIFT+CLR/HOME]
     QUIERES ESCRIBIR MAS
     DATAS (S/N)?"
1430 GET A$:IF A$="N" THEN
     170
1440 IF A$="S" THEN 1370
1450 GOTO 1430
```

EMPEZANDO

Al ejecutar el programa, el primer mensaje que aparezca te preguntará si quieres *sprites* multicolor o no. Si respondes S [RETURN] (por sí), aparecerá el primero de los tres mensajes de selección de color. El valor de color que introduzcas debe estar en el margen de 0 a 15, siguiendo la escala convencional del **Commodore 64** que empieza en el 0 con el negro. Puedes seleccionar tres colores, el cuarto color elegido corresponde al color de fondo.

Si pulsas N [RETURN], queda seleccionado el modo *sprite* de alta resolución en el que sólo recibes dos mensajes de control de colores. Por lo demás, las restantes rutinas son las mismas.

Cuando hayas seleccionado los colores, la pantalla presentará la cuadrícula de generación de *sprites*, un panel de información de estado (*status*) y una zona de presentación de sprites un poco más abajo. La cuadrícula tiene una resolución de 12 cuadrados horizontales en modo multicolor y 24 en alta resolución.

EDICION

Una vez que hayas hecho tu selección inicial de colores, el programa entra automáticamente en el modo de «edición» y puedes empezar a introducir tu dibujo en la retícula. Utiliza las teclas de cursor para desplazarte sobre la misma y las teclas 1, 2 y 3 para colorear una casilla.

Hay otros varios comandos de edición. Al pulsar [RETURN] el cursor se desplaza al principio de la siguiente línea, [CLR/HOME] lleva el cursor a la posición [HOME], [SPACE] borra el cuadrado anterior y [INST/DEL] borra el cuadrado que hay bajo el cursor.

El área de presentación de *sprites* va presentando el dibujo y el borrado a medida que se va produciendo, pudiendo ver en cualquier momento el aspecto que tiene. Puedes comparar su forma en modos normal y expandido haciendo uso de las teclas de funciones. Pulsando [F1] el *sprite* se expande horizontalmente, con F3 lo hace verticalmente. [F5] y [F7] hacen que el *sprite* recupere su tamaño normal en los ejes X e Y respectivamente. El programa empieza con un *sprite* expandido.

En cualquier momento durante el modo de edición puedes decidir cambiar el tipo de *sprite* o los colores, pulsa entonces [R] para recomenzar de nuevo.

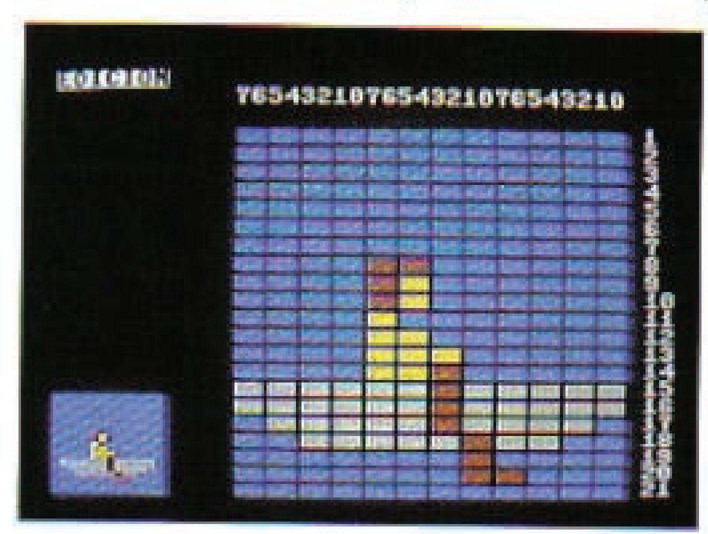
ALMACENAMIENTO DE LOS SPRITES

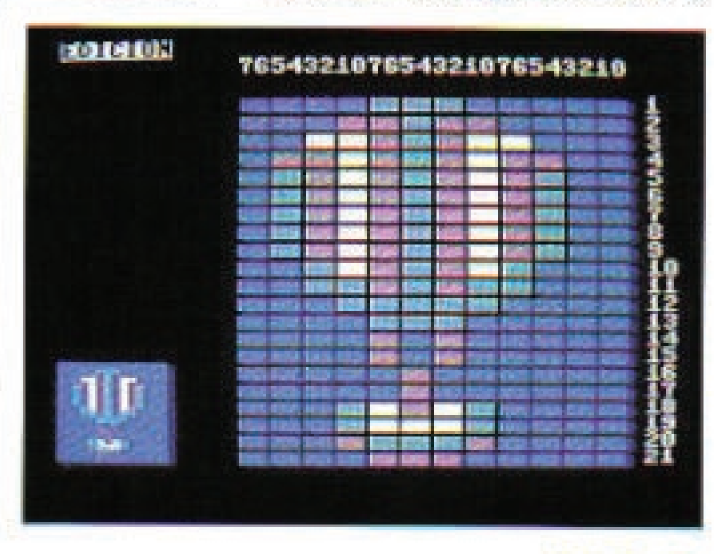
En cualquier momento, un *sprite* recién definido o modificado puede transferirse desde su situación de almacenamiento temporal en el *buffer* de la cinta (área de memoria que comienza en la dirección decimal 832), pulsando la tecla [S].

Para modificar un *sprite* ya existente, posiblemente debido a que lo utilices como punto de partida para otro *sprite*, pulsa la tecla [C]. Con ello el programa pasará a modo de «copia». El programa te pedirá que introduzcas un número de *sprite*. Vale cualquier número entre 0 y 63. El *sprite* elegido por tí se copia entonces desde la posición de memoria en la que esté almacenado a la pantalla, apareciendo tanto en la retícula como en el área de presentación de *sprites*.

Al utilizar el programa por primera vez, todas las direcciones de memoria de los *sprites* están llenas con cualquier cosa, pudiendo ocurrir que descubras así más de un *sprite* interesante. Prueba a seleccionar uno, por ejemplo el 56, y verás lo que sale.

Una vez el *sprite* en pantalla por medio del modo de copia, puedes trabajar sobre él o dejarlo para más adelante, trasladándolo de nuevo a la memoria con la tecla [S]. De nuevo se te





pedirá entonces otro número de *sprite* dentro del margen 1 a 63. Si quieres escribir encima del *sprite* utilizado como punto de partida para un nuevo diseño, teclea el mismo número que usaste para llamarlo. En caso contrario tendrás que elegir un nuevo número, pero fíjate que el *sprite* que ocupó la posición de memoria original quedará alterado durante este proceso, al escribirse el nuevo encima de él.

Cuando el *sprite* queda en la memoria, aparece una nueva información a la izquierda de la retícula. Son detalles sobre el puntero de ese *sprite*, en particular direcciones de comienzo y de final en la memoria del ordenador, de las que puedes tomar nota para posibles usos futuros o sacar una copia de impresora (como explicaremos más adelante).

OBSERVACION

Durante el modo de edición puedes visualizar en cualquier momento lo que hay en la memoria destinado a los *sprites*, pulsando la tecla [V] (de ver). El *sprite* sobre el que estabas trabajando sigue igual en la memoria, hasta que lo alteres mediante una instrucción de almacenamiento de los nuevos datos.

La visualización empieza por el *sprite* 0 y avanza muy rápidamente por toda la serie, pulsando la barra espaciadora, con lo que avanza un *sprite* cada vez. La facilidad con que puede hacerse esto ilustra la potencia de una simple modificación de los punteros del *sprite*. Para retroceder al *sprite* anterior, pulsa [B]. Si quieres usar dos *sprites* parecidos como base de una figura animada, puedes almacenarlos en dos posiciones contiguas y cambiar rápidamente de una a otra para ver el efecto que hacen.

Si te encuentras con un *sprite* que deseas revisar o utilizar como base para diseñar un nuevo *sprite*, utiliza la rutina de copia a la que llamas con la tecla [C]. En caso contrario, pulsa [RETURN] para entrar otra vez en el modo de edición.

SALIDA DE DATOS POR IMPRESORA

Si tienes conectada una impresora y está encendida, puedes sacar copia del número de *sprite*, puntero y direcciones de comienzo y final, pulsando la tecla [D], de datos, en el modo de visualización. El programa te brindará las opciones de pantalla e impresora.

Seguidamente puedes transferir los datos a otros programas introduciéndolos manualmente.

ENTRADA DE DATOS

Las definiciones de *sprites* a base de sentencias DATA con los datos en decimal se utilizan con frecuencia en los listados y también se pueden introducir directamente en el programa de generación de *sprites*. De esta forma puedes utilizar valores publicados en libros o revistas en forma de DATA como base para tus propios diseños. En este artículo veremos algunos ejemplos de ello; tendrás que transferirlos a la memoria, utilizando el procedimiento descrito en el siguiente párrafo.

Cuando estés en el modo de edición, pulsa [E] para hacer una entrada de datos. Primero se te pedirá el número de *sprite* que quieres asignar a tu nuevo diseño y a continuación los 64 grupos separados de valores de DATA. Cuando hayas terminado de introducir los datos, el programa te preguntará si quieres continuar. De no ser así, el programa regresa al modo de edición y en la pantalla aparecerá el último *sprite* que se introdujo, que también está almacenado en la memoria.

ALMACENAMIENTO DE LOS DATOS DEL SPRITE

Los datos que están en la memoria se pueden salvar en cualquier momento. Para hacerlo, pulsa [*] y saldrás del modo de edición. Aparecerá entonces un mensaje preguntándote si quieres guardar el último *sprite* sobre el que estabas trabajando. Se trata de una medida de precaución: responde con sí o no. Volverás al modo de edición (en el que puedes usar la función de almacenamiento) o se te preguntará si quieres hacer SAVE o LOAD. Si eliges la opción SAVE, el programa te presentará la opción una pantalla con los comandos directos predeterminados.

Para ejecutar la rutina de SAVE, simplemente lleva el cursor a la primera de las dos líneas, teclea un nombre correcto de *sprite* y pulsa [RETURN]. Realmente la línea está escrita para los usuarios de cinta: si quieres hacer el almacenamiento en disco, debes modificar el número de dispositivo de la línea 1340, cambiándolo por un 8.

PUEDES VER PUBLICADO TU SPRITE

Con toda seguridad diseñaréis magníficos *sprites* perfectamente utilizables en juegos. Enviándonos las listas con los DATA que los dibujan, publicaremos aquellos que puedan ser empleados por otros usuarios para desarrollar sus programas.

Escribir en el sobre: «SPRITES/COMMODORE».

BARAJA Y REPARTE

- DISPOSICON DE LAS CARTAS
- DIBUJANDO LAS CARTAS
- REPARTO
- CUATRO PALOS, TRECE CARTAS
- VALORES DE LOS ASES Y LAS FIGURAS

Los ordenadores pueden ser muy buenos jugadores de cartas si se programan correctamente; además nunca se aburren. Aquí tienes la forma de programar los gráficos de una baraja.

¿Te encuentras distanciado de tus amigos, parientes o colegas por haberles dejado sin un duro jugando a las cartas? ¿Eres tú el que estás sin blanca por haber jugado con expertos? Sea como fuere, en los tres capítulos siguientes te presentamos la solución. Programando tu ordenador para que juegue contigo a las veintiuna, tendrás una víctima propiciatoria y una manera de jugar sin tener que vaciarle los bolsillos a nadie.

En esta primera parte nos ocuparemos de la manera de generar las rutinas gráficas con las que se construye la baraja. El resto del programa, que es el juego propiamente dicho, se presentará en los dos capítulos siguientes. Pero no te olvides de almacenar cada una de las secciones en cinta a medida que vas construyendo el juego.

Si no eres un experto en el juego de las veintiuna, no te preocupes. En la última parte del programa presentaremos un conjunto completo de las reglas del juego. Pero antes debes ser capaz de programar un mazo de cartas.

El siguiente programa es la rutina de gráficos para el **Commodore**. Si tie-



nes un **Vic 20**, asegúrate de cambiar la línea 10, que se convertirá en la siguiente:

```
10 POKE 36879,30:MU=100:JM=3
   :LE=21:TX=6
```

Los poseedores de un **Commodore** estáis de suerte, ya que todos los símbolos necesarios para los gráficos de la baraja están incorporados en la máquina, por lo que no hay necesidad de escribir rutinas especiales de gráficos.

```
10 POKE 53280,6:POKE 53281,1
   :MU=100:JM=7:LE=39:TX=15
20 FOR Z=1 TO LE:E1$=E1$+
   "[COMM+P]":E2$=E2$+
   "[COMM+-]":NEXT Z:E2$=E2$
   +"[COMM+-]"
30 DIM D$(52):C$="[SHIFT+A]
   [SHIFT+S][SHIFT+X]
   [SHIFT+Z]":CC$="A23456789
   0JQK"
40 FOR ZZ=1 TO 4:FOR Z=1 TO
   13
50 X=X+1:D$(X)=MID$(C$,ZZ,1)
   +MID$(CC$,Z,1):NEXT Z,ZZ:
   PRINT "[SHIFT+CLR/HOME]"
60 FOR X=1 TO 52
70 XX=INT(RND(1)*52)+1
80 DD$=D$(X):D$(X)=D$(XX):
   D$(XX)=DD$:IF RND(1)<.50
   THEN 130
90 PRINT"[CLR/HOME][4*CRSR
   abajo]":T=RND(1)*3+TX
100 PRINT TAB(T)"[CTRL+6]
    [SHIFT+O][5*COMM+Y]
    [SHIFT+P]":FOR ZZ=1 TO 9
110 PRINT TAB(T)"[CTRL+6]
    [COMM+H][CTRL+7]
    [5*COMM++][CTRL+6]
    [COMM+N]":NEXT ZZ
120 PRINT TAB(T)"[SHIFT+L]
    [5*COMM+P][SHIFT+@]"
130 NEXT X
140 Z=0
150 PRINT"[SHIFT+CLR/HOME]
    [11*CRSR abajo]"
160 FOR ZZ=1 TO 10:PRINT
    "[CTRL+1][CTRL+9]";E1$:
    NEXT ZZ
170 PRINT"[CLR/HOME]
    [4*CRSR abajo]";:PRINT
    "[CTRL+7][CTRL+9]"E2$;
    E2$
180 Z=Z+1:NU=NU+1:IF Z>52
    THEN Z=1
190 DD$=LEFT$(D$(Z),1)
200 D1$=RIGHT$(D$(Z),1):IF
    D1$="0" THEN D1$="10":
    D2$="10":GOTO 220
```



```
210 D2$=" "+D1$:D1$=D1$+" "
220 FOR D=1 TO 500:NEXT D:
    PRINT"[CLR/HOME]CARTA
    [CTRL+7][3*ESPACIO]
    [3*CRSR dcha.]
240 PRINT"[CLR/HOME]
    [7*CRSR abajo][CTRL+6]"
    TAB(T)"[CTRL+9][SHIFT+O]
    [5*COMM+Y][SHIFT+P]":FOR
    ZZ=1 TO 9
250 PRINT TAB(T)"[CTRL+9]
    [COMM+H][CTRL+O]
    [5*ESPACIO][CTRL+9]
    [COMM+N]":NEXT ZZ
260 PRINT TAB(T)"[CTRL+9]
    [COMM+L][5*COMM+P]
    [SHIFT+@][CTRL+1]":IF
    DD$="[SHIFT+Z]"OR DD$=
    "[SHIFT+S]" THEN PRINT
    "[CTRL+3]"
270 PRINT "[CLR/HOME][8*CRSR
    abajo]" TAB(T);
    "[CRSR dcha]";D1$;
    "[8*CRSR abajo][CRSR
    dcha]";D2$
280 FOR ZZ=1 TO 13: IF MID$
    (CC$,ZZ,1)=RIGHT$(D$(Z),
    1) THEN JJ=ZZ
290 NEXT ZZ:PRINT"[CLR/HOME]
    [9*CRSR abajo]"TAB(T);
    "[CRSR dcha]";
300 ON JJ GOSUB 700,520,530,
    540,550,560,570,580,590,
    610,650,670,630
510 GO TO 180
520 PRINT"[CRSR abajo]
    [2*CRSR dcha]"DD$"[CRSR
    izq.][4*CRSR abajo]"DD$:
    RETURN
530 PRINT "[CRSR abajo]
    [2*CRSR dcha]"DD$"[CRSR
    izq.][2*CRSR abajo]"DD$"
    [CRSR izq.][2*CRSR abajo]
    "DD$:RETURN
540 PRINT"[CRSR abajo][CRSR
    dcha]"DD$"[CRSR dcha]"
    DD$"[3*CRSR izq.]
    4*[CRSR abajo]"DD$"[CRSR
    dcha]"DD$:RETURN
550 PRINT"[CRSR abajo]
    [CRSR dcha]"DD$"[CRSR
    dcha]"DD$"[2*CRSR izq.]
    [2*CRSR abajo]"DD$
    "[2*CRSR izq.][2*CRSR
    abajo]"DD$"[CRSR dcha]"
    DD$:RETURN
560 PRINT"[CRSR abajo][CRSR
    dcha]"DD$"[CRSR dcha]"
    DD$"[3*CRSR izq.][2*CRSR
    abajo]"DD$"[CRSR dcha]"
    DD$"[3*CRSR izq.][2*CRSR
    abajo]"DD$"[CRSR dcha]"
    DD$:RETURN
```



```
570 PRINT "[CRSR abajo][CRSR
    dcha]"DD$"[CRSR dcha]"
    DD$"[2*CRSR izq.][CRSR
    abajo]"DD$"[2*CRSR izq.]
    [CRSR abajo]"DD$"[CRSR
    dcha]"DD$"[3*CRSR izq.]
    [2*CRSR abajo]"DD$
    "[CRSR dcha]"DD$:RETURN
580 PRINT "[CRSR abajo][CRSR
    dcha]"DD$"[CRSR dcha]"
    DD$"[2*CRSR izq.][CRSR
    abajo]"DD$"[2*CRSR izq.]
    [CRSR abajo]"DD$"[CRSR
    dcha]"DD$"[2*CRSR izq.]
    [CRSR abajo]"DD$"[2*CRSR
    izq.][CRSR abajo]"DD$
    "[CRSR dcha]"DD$:RETURN
590 PRINT"[CRSR abajo][CRSR
    dcha]"DD$"[CRSR dcha]"
    DD$"[3*CRSR izq.][CRSR
    abajo]"DD$"[CRSR dcha]"
    DD$"[2*CRSR izq.][CRSR
    abajo]"DD$"[2*CRSR izq.]
    [CRSR abajo]"DD$"[CRSR
    dcha]"DD$"[3*CRSR izq.]
    [CRSR abajo]"DD$"[CRSR
    dcha]"DD$
600 RETURN
610 PRINT "[CRSR dcha]"DD$
    "[CRSR dcha]"DD$"[2*CRSR
    izq.][CRSR abajo]"DD$
    "[2*CRSR izq.][CRSR
    abajo]"DD$"[CRSR dcha]"
    DD$"[3*CRSR izq.][2*CRSR
    abajo]"DD$"[CRSR dcha]"
    DD$"[2*CRSR izq.][CRSR
    abajo]"DD$"[2*CRSR izq.]
    [CRSR abajo]";
620 PRINT DD$"[CRSR dcha]"
    DD$:RETURN
630 PRINT"[CTRL+9][5*COMM+*]
    [CTRL+0][CRSR abajo]
    [5*CRSR izq.][5*COMM++]
    [CRSR abajo][5*CRSR izq.]
    [COMM++][SHIFT+Q]
    [SHIFT+I][SHIFT+Q]
    [COMM++][CRSR abajo]
    [5*CRSR izq.][COMM++]
    [ESPACIO][COMM+Z]
    [ESPACIO][COMM++][CRSR
    abajo][5*CRSR izq.]
    [COMM++][SHIFT+J][SHIFT+
    C][COMM+K][COMM++][CRSR
    abajo][5*CRSR izq.]
    [SHIFT+J][COMM+R][SHIFT+
    C][COMM+R][COMM+K]";
640 PRINT"[CRSR abajo]
    [5*CRSR izq.][COMM+D]
    [CTRL+9][ESPACIO]"DD$
    "[ESPACIO][CTRL+0]
    [COMM+F]":RETURN
650 PRINT"][2*ESPACIO]
    [COMM+§][2*ESPACIO][CRSR
    abajo][5*CRSR izq.]
    [COMM+§][3*COMM++]
    [COMM+§][CRSR abajo]
    [5*CRSR izq.][COMM++]
    [SHIFT+Q][ESPACIO]
    [SHIFT+Q][COMM++][CRSR
    abajo][5*CRSR izq.]
    [COMM++][3*ESPACIO]
    [COMM++][CRSR abajo]
    [5*CRSR izq.][SHIFT+-]
    [SHIFT+J][SHIFT+C]
    [COMM+K][SHIFT+-][CRSR
    abajo][5*CRSR izq.]
    [SHIFT+J][COMM+R][SHIFT+
    C][COMM+R][COMM+K]";
660 PRINT"[CRSR abajo]
    [5*CRSR izq.][ESPACIO]
    [CTRL+9][ESPACIO]"DD$
    "[ESPACIO][CTRL+0]":
    RETURN
670 PRINT"[ESPACIO][CTRL+9]
    [COMM+B][ESPACIO]
    [CTRL+0][COMM+B]
    [ESPACIO][CRSR abajo]
    [5*CRSR izq.][CTRL+9]
    [2*COMM+B][CTRL+0]
    [ESPACIO][2*COMM+B][CRSR
    abajo][5*CRSR izq.][CTRL
    +9][COMM+B][CTRL+0]
    [SHIFT+Q][ESPACIO]
    [SHIFT+Q]";
680 PRINT"[COMM+B][CRSR
    abajo][5*CRSR izq.][CTRL
    +9][COMM+B][CTRL+0]
    [ESPACIO][SHIFT+W]
    [ESPACIO][COMM+B][CRSR
    abajo][5*CRSR izq.][CTRL
    +9][COMM+B][CTRL+0]
    [SHIFT+J][SHIFT+C][COMM+
    K][COMM+B][CRSR abajo]
    [5*CRSR izq.][CTRL+9]
    [COMM+B][CTRL+0][SHIFT+M]
    [ESPACIO][SHIFT+N]
    [COMM+B]";
690 PRINT"[CRSR abajo][5*CRSR
    izq.][ESPACIO][CTRL+9]
    [SHIFT+§]"DD$"[COMM+*]
    [CTRL+0]":RETURN
700 PRINT"[2*CRSR dcha]"DD$
    "[2*CRSR izq.]
    [CRSR abajo][COMM+§]
    [COMM++][COMM+§][CRSR
```



```
abajo][3*CRSR izq.]
[COMM++][ESPACIO][COMM++]
[CRSR abajo][3*CRSR izq.]
[COMM++][COMM+§][COMM++]
[CRSR abajo][3*CRSR izq.]
[COMM++][ESPACIO][COMM++]
[CRSR abajo][3*CRSR izq.]
[COMM++][ESPACIO][COMM++]
[CRSR abajo][2*CRSR izq.]
"DD$:RETURN
```

Los dos POKEs de la línea 10 del programa del **64** establecen los colores del fondo y borde de pantalla, mientras que el único POKE del programa del **Vic 20** se ocupa del color de la pantalla. MU son las piedras —que ahora bautizaremos como *chips*— para apostar, que el jugador necesitará más adelante durante el juego, JM se usa para visualizar las cartas en la pantalla y TX se utiliza para centrarlas durante la operación de barajado, que verás cuando ejecutes el programa. LE es la anchura de la pantalla del **Commodore** menos uno.

La línea 20 está relacionada con los gráficos utilizados como base para la visualización de las cartas; posteriormente se presentan en el programa por medio de las líneas 160 y 170.

En la línea 30 se dimensiona una matriz y se definen dos cadenas de caracteres. D$ contendrá las 52 cartas, C$ los 4 palos y CC$ los números y letras utilizados en las esquinas de los naipes.

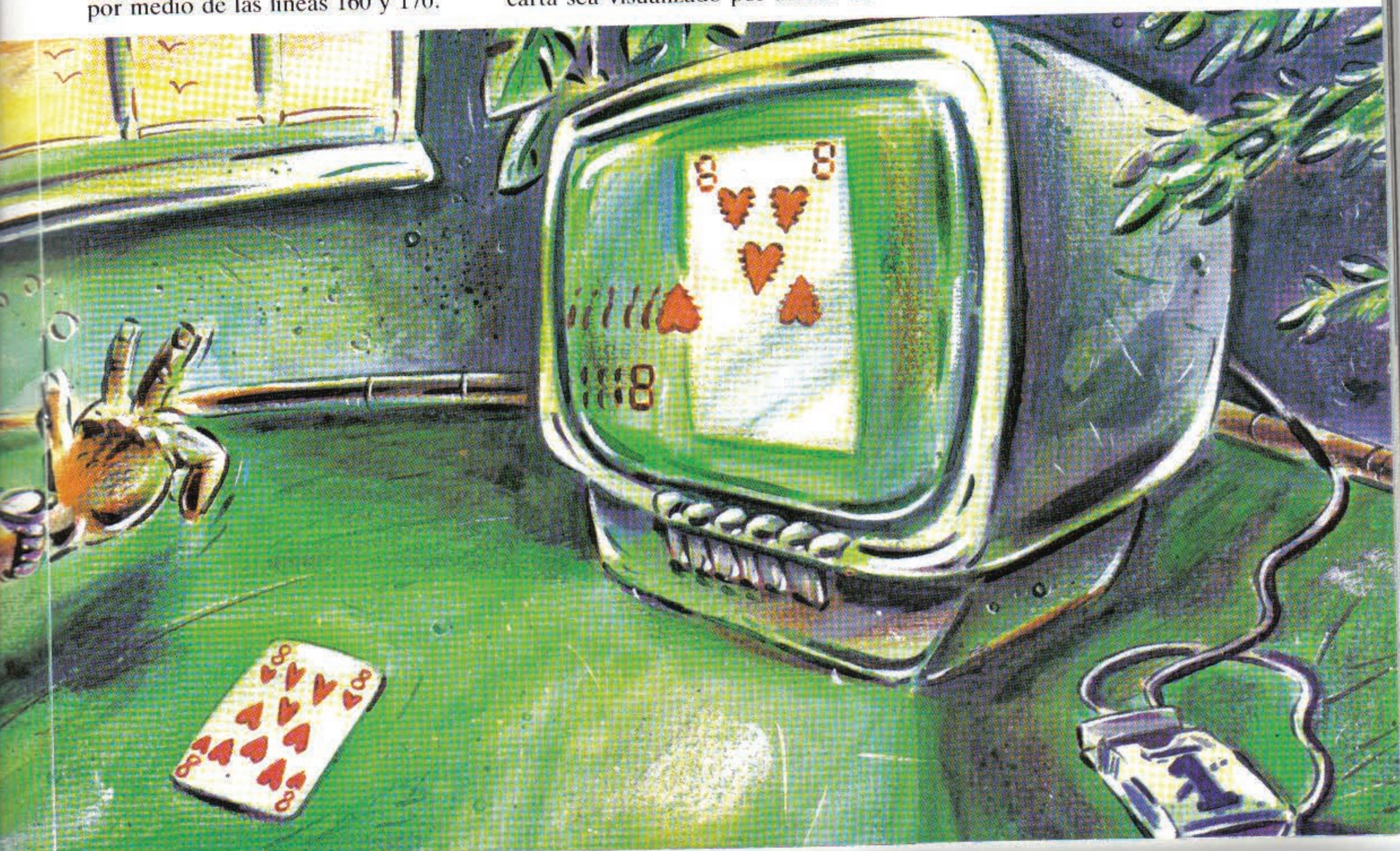
El mazo de cartas se define en las líneas 40 y 50. Ahora que ya ha sido creado y está en orden, hay que barajar, lo que se hace en las líneas 60 a 130. Las líneas 100 a 120 visualizan en pantalla las cartas que están siendo barajadas.

Una vez que las cartas están colocadas al azar por medio de la rutina de barajado, pueden ser repartidas en la pantalla. La línea 180 incrementa el número de carta dentro del mazo, Z, y el número de cartas repartidas hasta ese momento, NU. Esta línea también pone el contador de naipes al principio de la baraja cuando se llega a la última carta, es decir cuando se alcanza el número 52.

Las líneas 190 a 300 sirven para establecer la carta que queda encima del montón y visualizarla. El valor de la carta y el palo a que pertenece se calculan en las líneas 190 a 210. Hay una pausa antes de que el número de la carta sea visualizado por medio de la línea 220. La silueta de la carta se dibuja empezando siete líneas por debajo del borde de pantalla.

Ahora tienes un rectángulo blanco que es una carta sin vestir, en la cual hay que ir poniendo los símbolos del palo y los números o letras del valor. La línea 270 se ocupa de los números o letras, representando un carácter desde el as hasta el rey en las esquinas superior izquierda e inferior derecha del naipe.

Para poder visualizar en la pantalla el carácter gráfico adecuado, el programa tiene que decidir exactamente de qué carta se trata: ¿es un 3, un 8 o una dama? JJ es un número del 1 al 13 que representa desde el as hasta el rey y se utiliza para acceder a la línea correcta de símbolos gráficos. Estos símbolos gráficos están escritos en forma de subrutinas entre las líneas 510 y 700. Cada vez que aparece DD$ en estas líneas, se imprime en la pantalla el símbolo del correspondiente palo. Las líneas 510 a 600 se ocupan de la distribución de los dibujos y las 610 a 700 son las cabeceras de las cartas con figura, que se construyen a partir de una combinación de los símbolos gráficos del **Commodore**.



EMPIEZA EL JUEGO

- REPARTO DE LAS CARTAS
- LAS APUESTAS
- DOBLAR, CARTA Y PLANTARSE
- HACIENDO SALTAR A LA BANCA

El banquero fija en tí su mirada de hielo. Tú haces tu apuesta y recibes otra carta, pero ¿doblas, pides carta o te plantas? En esta ocasión nos ocuparemos de las líneas de programa correspondientes a la parte del jugador.

Continuando a partir de la rutina de gráficos que tecleaste en el capítulo anterior, hacen falta dos secciones más de programa: una para manejar las respuestas del jugador y la otra para hacer posible que pueda jugar el ordenador. Ahora veremos las líneas que hacen falta para el jugador. Pero no llegarás muy lejos en el juego, ya que todavía no has enseñado a jugar a tu ordenador.

Esta sección de programa está relacionada con unas cuantas tareas diferentes. (No te preocupes si no estás muy seguro de las reglas exactas del juego de las veintiuna, ya nos ocuparemos de ellas junto con la última parte del programa.

Básicamente el programa tiene que hacer tres cosas: Ocuparse del reparto de las cartas, permitir al jugador hacer apuestas y pedir cartas adicionales, y finalmente, tiene que calcular la puntuación del jugador.

En la siguiente sección de programa, se le pide al jugador que haga una apuesta sobre la primera carta que reparta el ordenador (la banca), una para el jugador y otra para el propio ordenador. Después de recibir la segunda carta, se le presentan tres opciones al jugador (que no veremos en pantalla aún): plantarse, doblar o pedir carta, según el estado del juego.

El programa tiene rutinas para manejar las tres opciones, repartiendo cartas extras y/o añadiéndolas en la puesta, o pasar la vez al banquero. Después de cada nueva carta el programa comprueba si la puntuación ha pasado de 21, en cuyo caso el jugador pierde.

El programa comprueba además las condiciones especiales: las veintiuna y



el juego de cinco cartas. Si se produce alguna de ellas, aparece un mensaje que dice al jugador la puntuación obtenida y pasa el turno al banco.

Añádele las siguientes líneas al programa que introdujiste anteriormente. El jugador estará en condiciones de practicar la mitad del juego, pero el ordenador no podrá replicarle hasta más adelante.

Teclea para Commodore-64

```
140 Z=0:GOTO 760
220 T=T+JM
230 FOR C=0 TO 159:POKE 1024
    +C,32:NEXT C
310 POKE 198,0
320 IF JJ>9  THEN TU=TU+10
330 IF JJ<10 THEN TU=TU+JJ:
    IF JJ=1 THEN T2=1
340 PRINT "[CLR/HOME]
    [CTRL+7]";S2$;" TOTAL:";
    TU;:T3=0:IFT2=1THENT3=10
    :IFTU+T3>21 THEN T3=0
350 IF T2>0 AND TU+10<22
    THEN PRINT "O";TU+10
360 IF TU+T3=21 AND NU=2
    THEN PRINT"[CLR/HOME]
    [CRSR abajo][CTRL+3]
    PONTON!":PO(PL)=1:GOTO
    800
370 IF TU>21 THEN PRINT
    "[CLR/HOME][CRSR abajo]
    [CTRL+3]PIERD";S1$:GOTO
    800
380 IF NU=5 THEN PRINT
    "[CLR/HOME][CRSR abajo]
    [CTRL+3]JUEGO DE CINCO
    CARTAS!": CT(PL)=1: GOTO
    800
400 IF TU=21 THEN PRINT
    "[CLR/HOME][CRSR abajo]
    [CTRL+3]TIENES 21!":GOTO
    800
420 IF PL=1 AND NU=1THEN 970
430 K1$="C":K2$="C":PRINT
    "[CLR/HOME][CRSR abajo]
    [CTRL+5]";:IFEB=0AND MU>
    =BE THENPRINT"(D)OBLAR "
    ;:K1$="D"
440 PRINT"(C)ARTA ";
450 IF TU>15 OR TU+T3>15
    THEN PRINT "(P)LANTE";:
    K2$="P"
460 PRINT "?":POKE 198,0
470 GET K$:IF K$<>K1$ AND
    K$<>"C" AND K$<>K2$ THEN
    470
480 IF K$="P" THEN PRINT
    "[CLR/HOME][2*CRSR abajo]
    [CTRL+6]BIEN.":GOTO 800
490 IF K$="C" THEN EB=1
500 IF K$="D" THEN TB=TB+BE:
    MU=MU-BE
760 TB=0:EB=0:FOR PL=1 TO 2
770 IF PL=1 THEN S1$="ES TU"
    :S2$="TU"
790 TU=0:NU=0:T=-JM:T2=0:T3=
    0:PO(PL)=0:CT(PL)=0:GOTO
    150
800 RUN
970 CH$="CHIPS":IF MU=1 THEN
    CH$="CHIP"
980 PRINT"[CLR/HOME]
    [CRSR abajo][CTRL+6]";
    CH$;":[CTRL+3]";MU
990 BE=0:INPUT"[CLR/HOME]
    [2*CRSR abajo][CTRL+5]
    APUESTA INICIAL ";BE:IF
    BE>MU OR BE=0 THEN 990
1000 TB=BE:MU=MU-BE:GOTO 180
```



Si tienes un **Vic 20**, tienes que asegurarte de que la línea 230 se escriba como sigue para que la pantalla quede bien ajustada:

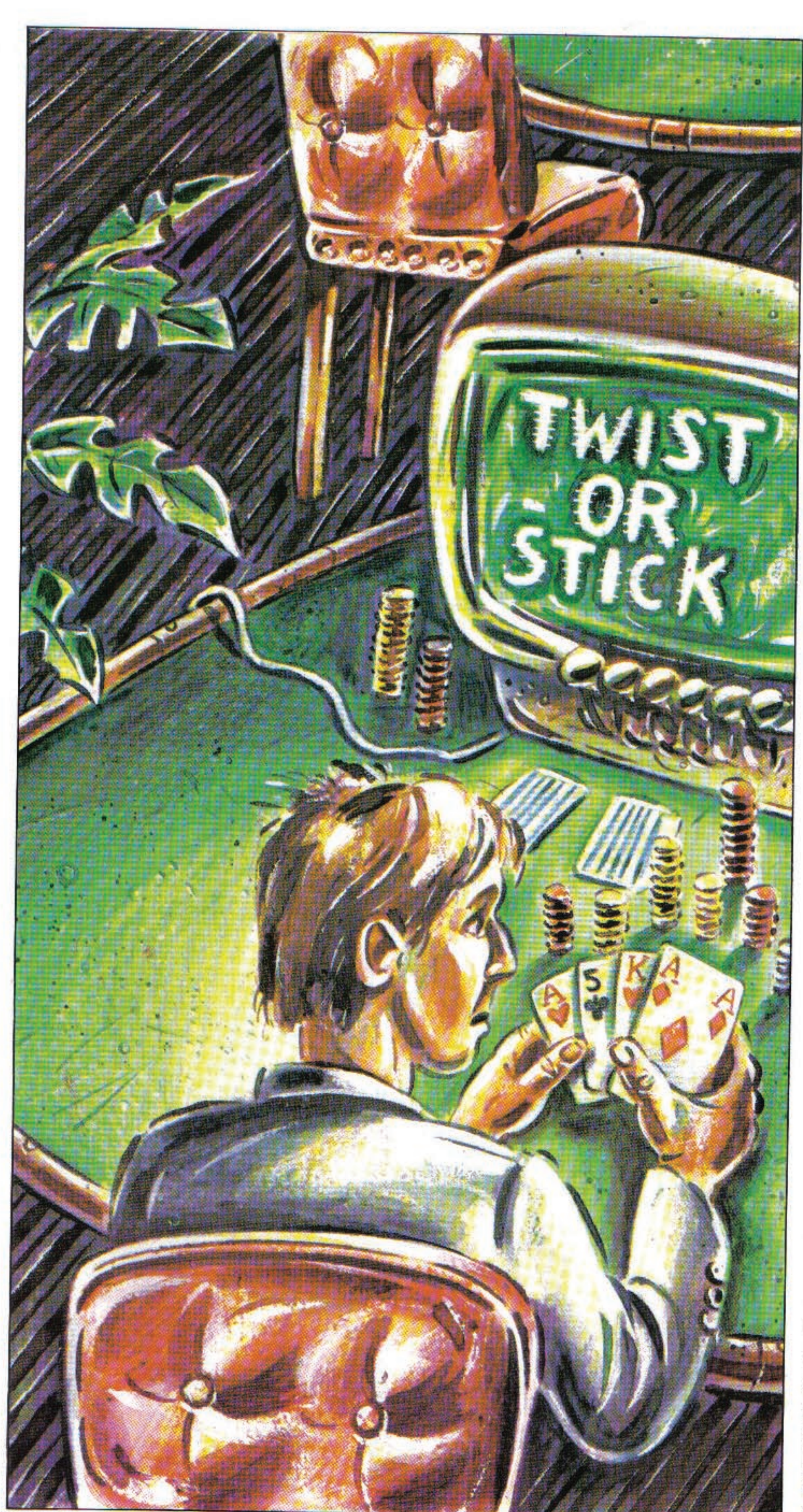
Teclea para Vic-20

```
230 FOR C=0 TO 87:POKE 7680,
    C+32:NEXTC
```

La línea 140 pone Z a cero. Con esto se asegura que el programa empieza repartiendo cartas por la parte superior del mazo. El programa salta a la línea 760 que pone a cero la apuesta total TB, así como el indicador de cambio, EB. Además comienza aquí un bucle FOR ... NEXT. Tal como está el programa ahora, puede parecer un poco peculiar, ya que todavía no se le ha añadido la instrucción NEXT para cerrar el bucle. No es una práctica de programación muy buena, pero recuerda que el programa está todavía siendo desarrollado.

La línea 770 define S1$ y S2$ para el jugador uno. El total y el número de cartas se ponen a cero, quedando listos para que empiece la mano. A la variable T se le asigna el valor −JM, de forma que la representación aparezca correctamente en pantalla. T2 y T3 se ponen a cero; cuando se reparte un as, T2 se pone a 1, como indicador y T3 a 10, como corrección para el caso en que se elige el valor 11 para el as. También se ponen a cero PO y CT en esta fase. El programa vuelve ahora a la línea 150 en la que se reparten las cartas.

La línea 220 calcula la posición de la siguiente carta, antes de que la lí-



nea 230 borre las cuatro líneas superiores de cualquier texto que haya sido presentado. El *buffer* del teclado se borra con el POKE de la línea 310.

Si la carta es un diez o una figura, en la línea 320 se le suma 10 al total actual. Si la carta no es un diez ni una figura, la línea 330 suma el valor de la carta al total actual. Si la carta es un as, T2 se pone a 1.

TOTALES

El valor total se representa en la línea 340. Si hay un as presente, se suma diez al total actual. Si dicho total es mayor que 21, se ignora, pero si no lo es, la línea 350 se ocupa de presentar la segunda puntuación. La línea 360 comprueba si se han hecho las veintiuna y activa el indicador de veintiuna, PO si así ocurre. Análogamente, las líneas 370 a 400 comprueban si el jugador se ha pasado, si ha conseguido un triunfo de cinco cartas o las veintiuna, y envía un mensaje.

Las líneas 430 a 460 están relacionadas con las ofertas que recibe el jugador de las opciones de doblar, pedir carta o plantarse. Si el jugador no ha pedido y tiene un total por debajo de 21, se le presentan las opciones de doblar o pedir carta; si la puntuación es 16 o mayor, también se le ofrece la opción de plantarse. Si en algún momento el jugador ha decidido pedir carta, no se le ofrece la opción de doblar una carta.

Las líneas 470 a 500 se ocupan de manejar la respuesta a las opciones presentadas. Si decide doblar, la línea 500 cambia el total de *chips*.

APUESTAS

Las líneas 970 a 1000 son las que manejan las apuestas. Las líneas 970 y 980 representan el número de *chips* que quedan. La línea 990 invita al jugador a que haga una apuesta con la primera carta, comprobando si hay suficientes fondos para dicha apuesta. La línea 1000, hace el valor de TB igual al de la apuesta, restando dicho valor del número total de *chips*.

MUSICA Y ENVOLVENTES EN EL COMMODORE

- MODULACION DE UNA ONDA
- LA ENVOLVENTE DE AMPLITUD
- LA ENVOLVENTE DE ALTURA
- FUNCIONAMIENTO
- INTERDEPENDENCIA

Cuando domines los comandos de envolventes de tu ordenador estarás en condiciones de imitar una gran cantidad de sonidos y de músicas, que te resultarán muy útiles en tus otros programas.

La incorporación del sonido puede marcar la diferencia entre un programa del montón y otro que resulte interesante, atractivo y agradable de usar. En casi todos los ordenadores se pueden programar notas puras o ruidos para aplicaciones tales como juegos, melodías sencillas y efectos especiales. Sin embargo para aplicaciones más sofisticadas sería mucho mejor que pudieras modificar con tu micro los tonos generados para poder imitar sonidos: Desde una sirena de alarma hasta el gorjeo de un pájaro, pasando por un determinado instrumento musical. El BASIC te brinda directamente esta posibilidad por medio de los comandos de envolvente, que tienes disponibles de forma indirecta, POKEando posiciones de memoria.

¿QUE ES LA ENVOLVENTE DE UN SONIDO?

Los sonidos se producen en unos circuitos electrónicos llamados osciladores. Dichos circuitos generan una señal de una determinada frecuencia (que es la que fija el tono del sonido) y con una cierta amplitud (de la que depende el volumen sonoro). Cuando esta señal eléctrica ataca a un altavoz se emite una nota de esa frecuencia y con esa amplitud. Para cambiar las características de una nota, puedes modificar cualquiera de estos dos parámetros; con mayores frecuencias tendrás tonos más altos, con mayores amplitudes el sonido se escuchará más fuerte. No todos los ordenadores te permiten tanto control, por ejemplo el **Spectrum** sólo te permite variar la frecuencia (y la duración), pero no el volumen del sonido.

Al cambiar estos parámetros se genera un amplio margen de notas musicales. Si combinas todas las notas al mismo tiempo, te resultará un ruido de tono indeterminado, al que se lla-



ma ruido blanco. Con este método sólo se producen efectos sonoros bastante limitados. Ello se debe a que la nota producida no se adapta a los efectos característicos de un determinado instrumento musical; esto es por ejemplo lo que diferencia al sonido de la nota Do central ejecutada en un saxofón o en un piano. Para imitar por ejemplo el sonido de un piano o un órgano, las señales generadas por los osciladores del micro han de ser modificadas, es decir hay que conformarlas. Esto es precisamente lo que hace un sintetizador, lo cual le permite producir una amplia variedad de sonidos.

La conformación de una onda sonora es una forma de modulación; se trata del mismo principio que hace posible la transmisión de la voz por medio de las ondas de radio. En la radio hay una onda de una determinada frecuencia y amplitud (la onda portadora) sobre la que se superpone la señal de voz en forma de envolvente de la onda portadora sinusoidal. Si dicha envolvente modifica los valores de pico de la onda portadora, la amplitud ya no es constante sino que varía con arreglo a la señal de voz; a esto es a lo que se llama modulación de amplitud o AM. Si la envolvente modifica la frecuencia de la onda portadora, aumentándola o disminuyéndola con arreglo a las variaciones de la señal de voz, el efecto resultante es la modulación de frecuencia o FM.

De una forma parecida, los micros pueden tener envolventes de amplitud, como es el caso del **Commodore 64**, envolventes de frecuencia o ambas, aunque esto último es bastante infrecuente. El efecto de cada tipo de modulación consiste en la superposición de una calidad nueva y sutilmente diferente sobre la nota pura.

Por muy sofisticado que sea un micro, es muy poco probable que su capacidad de producir sonidos sea tan buena como la de un sintetizador, que es un dispositivo específicamente diseñado para la generación de sonidos. Así pues, es difícil que un ordenador pueda realizar una síntesis convincente de los sonidos habitualmente producidos por instrumentos acústicos. Sin embargo, utilizando con imaginación las envolventes podrás producir una amplia variedad de sonidos, algunos de los cuales son una razonable imitación de los instrumentos reales y otros son completamente diferentes de cualquier instrumento existente, pese a lo cual resultan útiles e interesantes.

LA ENVOLVENTE DE AMPLITUD

La forma más fácil de entender la envolvente de amplitud es ayudándose de un gráfico de volumen o amplitud en función del tiempo. Cuando un instrumento musical ejecuta una nota, se está aplicando energía a una cuerda, lámina o lo que sea, haciendo que empiece a vibrar y ponga en movimiento el aire que le rodea. Por ejemplo, el sonido de un órgano sube rápidamente hasta un valor máximo del



COMPULAND
PRIMERA TIENDA ESPECIALIZADA EN COMMODORE

Por fin en casa

DESPUES DE UNOS MUY CONTINUADOS VIAJES A LAS MAS IMPORTANTES FERIAS DE NUESTRA COMUNIDAD,NOS VEMOS EN CONDICIONES DE OFRECEROS TODA UNA AMPLIA GAMA DE PRODUCTOS QUE HARAN QUE SALTEIS DE ALEGRIA.

EN DORMUND Y LONDRES VIMOS TODA UNA SERIE DE ARTICULOS,QUE CALLARIAN AL COMMODORIANO MAS EXIGENTE.

GRACIAS A LAS GESTIONES REALIZADAS EN AMBOS PAISES,PODEMOS DECIR CON ORGULLO QUE HEMOS CONSEGIDO COSAS QUE TARDARAN MUCHO EN LLEGAR A ESPANA,SI ES QUE LLEGAN.

EN CUESTION DE SOFT HEMOS SUSCRITO UN COMPROMISO CON LAS CASAS PARA TENER LOS PROGRAMAS AL MISMO TIEMPO.

EMPECEMOS A ESPECIFICAR.

Club Software

PRIMERO ACLARAR QUE SE PUEDE PERTENECER AL CLUB DESDE CUALQUIER PUNTO DE ESPANA,HACIENDOSE LOS CAMBIOS Y ENVIOS POR CORREO,CARGANDO 300 PTS POR GASTOS DE ENVIO.

PARA LOS USUARIOS DEL MISMO COMO LO PROMETIDO ES DEUDA,EMPIEZAN LAS VENTAJAS.

1-CON LA COMPRA DE 3 PROGAMAS DEL MISMO PRECIO PAGARAN SOLAMENTE 2

2-SE EMPEZARA A CREAR UNA REVISTA DONDE PODEIS PARTICIPAR Y DAR VUESTRAS OPINIONES,AL MISMO TIEMPO SERVIRA DE CATALOGO

3-TAMBIEN OS RECUERDO EL 10% DEL QUE DISFRUTAIS EN TODOS LOS PRODUCTOS.

4-LA CUESTION MODEM TAMBIEN SE VA ACERCANDO Y ESPERAMOS QUE EN POCO TIEMPO PODAMOS EMPEZAR A DISFRUTAR DE EL.

Cartuchos

POR FIN NOS VEMOS EN CONDICIONES DE OFRECEROS TODA UNA SERIE DE CARTUCHOS INCLUSO DESCONOCIDOS PARA ALGUNOS USUARIOS.

LOS MAS CONOCIDOS:

NEW FINAL CARTRIDGE-	13800
FREEZE FRAME-	13800
ISEPIC-	16800
QUICKDISC PLUS-	7800
SIMON'S BASIC-	16800

FINAL CARTRIDGE,EXPANSOR DE CARTUCHOS, AMPLIACIONES DE MEMORIA,GRABADORES DE EPROM,ADMINISTRACION DE FINCAS Y OTROS MUCHOS MAS CAPACES DE PONER LOS DIENTES LARGOS AL MAS DURO COMMODORIANO. NO A LOS LARGOS TIEMPOS DE ESPERA Y A LA INCERTIDUMBRE DE PERDER EL DINERO.

Perifericos

AHORA ESTAMOS EN CONDICIONES DE OFRECEROS UNA AMPLIA GAMA DE PERIFERICOS EN UNA INTERESANTE RELACION PRESTACIONES PRECIO.

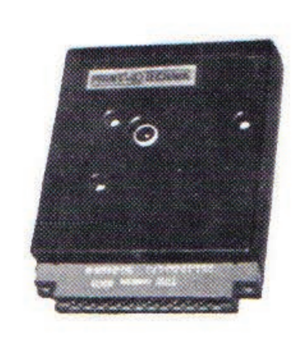
DIGITALIZADORES DE SONIDO:
-SOUND SAMPLER:APROXIMADAMENTE 20000
-SOUND EXPANDER: " 22000
DIGITALIZADORES DE VIDEO: 30000
-TANTO EN MODO 64 COMO 128
-PROXIMAMENTE PARA EL AMIGA
INTERFACES PARA COPIAS DE SEGURIDAD DE CASSETTE 4800
RATONES GRAFICOS ENTRE 15000 Y 20000
LAPICES OPTICOS ENTRE 5000 Y 10000
TABLETAS GRAFICAS ENTRE 15000 UY 20000
GAMES KILLER APROXIMADAMENTE 4000
CABLES MONITOR 40/80 COLUMNAS
Y UN AMPLIO CATALOGO.

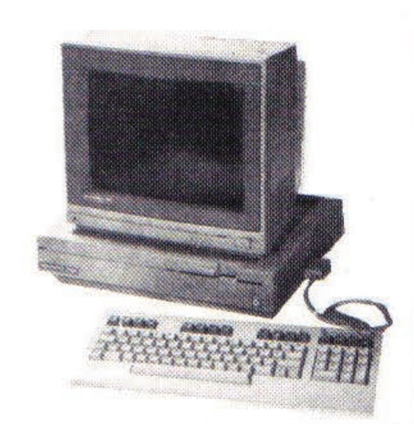
Ordenadores

EN ESTE APARTADO ESTAMOS MUY CONTENTOS POR PODEROS OFRECER LOS MEJORES PRECIOS DEL MERCADO,Y EN LA PRACTICA TOTALIDAD DE LA GAMA COMMODORE Y COMPATIBLES.

CBM 128 -------	65000
CBM 128D ------	130000
CBM 64 -------	44500
MPS 801 ------	33900
CANON COL.----	74000
128+1570------	105000
1571 --------	65000
MPS1000-------	58000
MPS803--------	40000
1901----------	70000
EPSONLX80-----	65000

PREGUNTE POR NUESTROS PAQUETES INTEGRADOS SE ASOMBRARA DE LOS PRECIOS.





COMPULAND

te acerca a tu casa los internacionales productos COMMODORE

Utilidades

EN UTILIDADES NOS LLEVAMOS EL PREMIO POR FIN DISPONIBLE EL FAMOSO LOTUS 1,2,3. LLAMADO VIZASTAR EN COMMODORE Y PRACTICAMENTE AL PRECIO INGLES.

-VIZASTAR 128 -------- 35000 (MODO 128)
-VIZAWRITE 128 -------- 29000 (MODO 128)
-VIZASTAR 64 --------- (LLAMAR)
-VIZAWRITE 64 -------- (LLAMAR)
-PAQUETES INTEGRADOS 128
- " " 64
-PRINTSHOP --------- 16000
-DOODLE --------- 16000
-NEWSROOM -------- 16000
-ALINEADOR 1541------ 13200
-LOGO ----------- 10000
-PASCAL ---------- 10000

Y UN CONJUNTO DE MAS DE 20 DISQUETTES PARA EL 128 EN MODO 128.

Concurso

DADA LA GRAN CONFIANZA QUE TENGO EN MIS COLEGAS COMMODORIANOS VAMOS A INICIAR UN CONCURSO DE SOFT,TANTO DE UTILIDADES COMO DE JUEGOS.

LOS MEJORES PROGRAMAS SERAN COMERCIALIZADOS Y SU AUTOR PARTICIPARA DE LOS BENEFICIOS.DANDO LA POSIBILIDAD A NUESTROS QUERIDOS COMPANEROS A QUE SU TRABAJO SE VEA RECOMPENSADO.

CONDICIONES:

-QUE NO SEAN PIRATEADOS,NI ENVIADOS A OTROS CONCURSOS.

-EN DISCO O EN CINTA Y A SER POSIBLE UN LISTADO APARTE POR IMPRESORA.

Y NO VAMOS A PPONER NINGUNA CONDICION MAS,SOLO DECIROS QUE LOS DE BUENA CALIDAD ENTRARAN EN LA NUEVA LINEA "ANTI-HACKERS" QUE VAMOS A COMERCIALIZAR.

Nuevos Productos

SIGUIENDO LA LINEA INICIADA POR NUESTROS COMPANEROS Y AMIGUETES DE DROSOFT,EMPEZAREMOS A DISTRIBUIR UNA SERIE DE JUEGOS DE MUY ALTA CALIDAD Y BAJO PRECIO QUE SI NUESTROS PROGAMADORES QUIEREN SE VERA INCREMENTADA CON SUS TRABAJOS,Y ESPERAMOS SEA DEL AGRADO DE TODOS VOSOTROS.

TAMBIEN EN BREVE EMPEZAREMOS A INSTALAR EL FAMOSO E INCREIBLE "SPEEDDOS" QUE LLEGA A ACELERAR LA VELOCIDAD DE CARGA EN 25 VECES,Y APARTE LLEVA INCORPORADO UN MONITOR DE CODIGO MAQUINA, PROGRAMA LAS TECLAS DE FUNCION Y OTRAS COSILLAS MAS.

OS RECUERDO VARIAS COSAS:

-EL DESCUENTO DEL 10% EN TODOS LOS PRODUCTOS A LOS SOCIOS DEL CLUB
-QUE SE ENVIA A TODA ESPAÑA
-SI OS INTERESA ALGO,OS LO CONSEGUIREMOS.

Rebajas

HAY UNA SERIE DE ARTICULOS EN LOS QUE LA CONTINUA LUCHA HA REBAJADO EL PRECIO.

-DISCOS VIRGENES:
SSDD-------300 PTS
DDDD-------325 PTS
-TALADRO DISQUETTES 1950
-TAPA ABATIBLE TRANSPARENTE 2350
-QUICKSHOT II 1950
-LAPIZ OPTICO 9650
-TABLETAS Y MUCHOS OTROS PRODUCTOS LLAMANOS Y PIDE INFORMACION SIN NINGUN COMPROMISO NO MOLESTAS.



COMPULAND
C/CALVO ASENSIO N.8
MADRID 28015
TFN:2431638
ARGUELLES

SI ESTAS INTERESADO EN CUALQUIER PRODUCTO LLAMANOS Y HAREMOS TODO LO POSIBLE POR CONSEGUIRLO.

SI BUSCAS ALGO DE COMMODORE VEN A COMPULAND,TU TIENDA.

COMPULAND NUMERO UNO EN COMMODORE.

volumen, el cual se mantiene a un nivel constante mientras la tecla se mantiene pulsada, extinguiéndose posteriormente. Por el contrario, en un piano o una guitarra la subida del volumen de la nota en el momento inicial es semejante al caso del órgano, pero inmediatamente el volumen empieza a disminuir y a extinguirse. Si se suelta entonces la tecla (o se toca la cuerda en el caso de la guitarra), el sonido resulta rápidamente silenciado. Para simular estas secuencias sobre la forma en que se ataca una nota, se mantiene y posteriormente se extingue, puedes programar una envolvente.

LA ENVOLVENTE DE ALTURA

La envolvente de altura resulta algo más difícil de entender que la envolvente de amplitud. También en este caso se puede representar en forma de gráfico en el que la frecuencia se acelera o se retrasa en función del tiempo, elevando o bajando el tono de la nota. Sin embargo la conformación de la envolvente no se produce de una forma tan sencilla como cambiar la altura de la nota a un valor distinto, ya que esta variación se produce de una forma controlada y regular dentro de un período.

Un ejemplo de cómo sería esto en un trozo musical sería el de un vibrato aplicado a una nota, lo que produce una especie de pulsación. O en materia de efectos sonoros podría ser la nota producida por una sirena cambiando hacia arriba y hacia abajo a un ritmo regular. Otro ejemplo muy corriente es el **efecto Doppler**, en el que el sonido procedente de un vehículo en rápido movimiento parece aumentar su altura cuando éste se acerca al que lo está escuchando, disminuyendo de nuevo al alejarse.

DISEÑANDO UN SONIDO

Aunque enseguida puedes aprender a analizar determinados sonidos, no siempre es fácil diseñar una envolvente que te produzca el efecto sonoro deseado, por lo que se trata principalmente de una cuestión de probar una vez y otra. He aquí un programa que te permite modificar los parámetros de la envolvente, escuchando posteriormente cada sonido:

Teclea para Commodore 64

```
10 POKE 650,128:POKE 53280,3
   :POKE 53281,3
20 A=4:D=7:S=6:R=9:FF=17
30 FOR I=1 TO 20:S$=S$+
   "[COMM++][CRSR arriba]
   [CRSR izq]":NEXT
40 RI$="[41*CRSR dcha]"
50 BOT$="[CLR/HOME][22*CRSR
   abajo]"
60 GOSUB 1000
70 POKE 54296,9
80 POKE 54273,40
100 GET X$
110 IF X$=" " THEN GOSUB 500
```



```
120 IF X$<>"" THEN GOSUB
     1000
130 GOTO 100
500 POKE 54276,FF
510 FOR I=1 TO 230:NEXT
520 GET X$:IF X$=" " THEN
     510
530 POKE 54276,FF-1
540 RETURN
1000 IF X$="A" THEN A=A+1
1010 IF X$="D" THEN D=D+1
1020 IF X$="S" THEN S=S+1
1030 IF X$="R" THEN R=R+1
1040 IF X$="[SHIFT+A]" THEN
      A=A-1
1050 IF X$="[SHIFT+D]" THEN
      D=D-1
1060 IF X$="[SHIFT+S]" THEN
      S=S-1
1070 IF X$="[SHIFT+R]" THEN
      R=R-1
1076 IF X$="[f1]" THEN FF=17
1077 IF X$="[f3]" THEN FF=33
1078 IF X$="[f5]" THEN FF=65
1079 IF X$="[f7]" THEN
      FF=129
1080 A=A AND 15:D=D AND 15:
      S=S AND 15:R=R AND 15
1090 POKE 54277,A*16+D
1100 POKE 54278,S*16+R
1120 PRINT"sQQQRATA="A,"DEC
      ="D,"SOS="S,"REL="R
1130 PRINT"[CLR/HOME][CTRL+
      7]PULSA [CTRL+6]LA
      TECLA[CTRL+7] PARA
      INCREMENTAR VALOR"
1140 PRINT"PULSA [CTRL+6]
      SHIFT[CTRL+7] Y[CTRL+6]
      LA TECLA[CTRL+7] PARA
      REDUCIRLO"
1150 PRINT"PULSA [CTRL+6]
      ESPACIO[CTRL+7] PARA
      SONIDO"
1200 XX=0:HT=0
1210 HT=HT+(16-A)/4:IF HT>20
      THEN HT=20
1220 GOSUB 2000
1230 IF(HT<20) AND (XX<40)
      THEN 1210
1240 IF XX=40 THEN RETURN
1250 HT=HT-(16-D)/4:IF HT<20
      *S/15 THEN HT=20*S/15
1255 IF HT<1 THEN HT=0
1260 GOSUB 2000
1270 IF(HT>20*S/15) AND (XX<
      40) THEN 1250
1280 IF XX=40 THEN RETURN
1290 FOR X=XX TO 30:REM FASE
      SOSTENIDA
1300 GOSUB 2000
1310 NEXT
1320 HT=HT-(16-R)/4:IF HT<0
      THEN HT=0
1330 GOSUB 2000
1340 IF(HT>0) AND (XX<40)
      THEN 1320
1350 RETURN
2000 PRINT BOT$
2005 IF HT=0 THEN RETURN
2010 PRINT LEFT$(RI$,XX);
2020 PRINT LEFT$(S$,HT*3+1)
2030 XX=XX+1
2040 RETURN
```

El **Commodore** sólo te permite conformar la envolvente de amplitud de la onda básica, pudiendo comenzar a partir de cuatro tipos de ondas. A continuación, para sintetizar un tipo particular de sonidos puedes programar cuatro fases: ataque (A), decaimiemto (D), sostenimiento (S) y relajamiento (R). A este tipo de envolventes se le llama algunas veces sistema ADSR. La fase de ataque determina la velocidad a la que crece el volumen cuando se inicia el sonido. La fase de caída fija el momento inicial de la extinción, viniendo posteriormente el sostenimiento, que es el nivel que se mantiene estable, como en el caso de un órgano. La cuarta fase —relajamiento— es la velocidad a la que el volumen cae a cero.

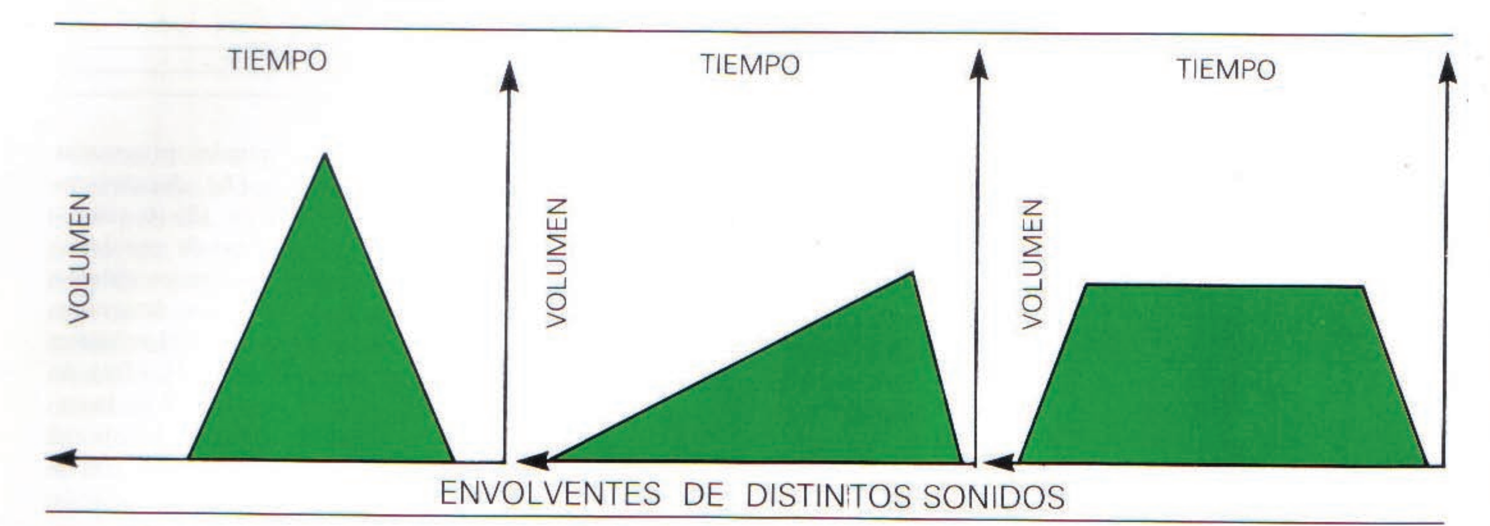
Ejecuta el programa para ver los valores iniciales de ADSR que aparecen en la parte superior de la pantalla, así como un gráfico de la envolvente producida por estos valores. Para tener una nota, pulsa la barra espaciadora.

Puedes cambiar los valores de ADSR de este programa pulsando las teclas adecuadas: [A], [D], [S] o [R] para aumentar los correspondientes valores y SHIFT con las mismas teclas para disminuir dichos valores. Después de cada pulsación de tecla, se actualizarán sobre la pantalla los valores de ADSR y el gráfico.

El tipo de sonido producido por cualquier conjunto de valores de ADSR depende de la forma de onda básica sobre la que actúa la envolvente. El programa te permite cambiar de una a otra entre las cuatro disponibles utilizando las teclas de funciones. Con la tecla [F1] tendrás un triángulo, con la [F3] un diente de sierra, con la [F5] una señal pulsada y con la [F7] ruido. Para determinados juegos de valores, podría ser que no te resultara ningún sonido al pulsar [F5], debido a que el sonido empiece a decaer antes de haber alcanzado un valor audible.

Cuando vayas a programar tus propios sonidos, es útil entender la forma





ENVOLVENTES DE DISTINTOS SONIDOS

en que funciona este programa, aunque en la práctica sólo necesitarás unas cuantas líneas de programa para conseguir sonidos bastante complicados. La línea 10 activa la autorrepetición de las teclas, para detectar cuándo se mantiene pulsada la barra espaciadora. La línea 20 establece los valores iniciales de A, D, S y R. La línea 30 define una secuencia de caracteres para dibujar el gráfico (un histograma) de la envolvente. La línea 40 define una secuencia para desplazar el cursor hasta el fondo de la pantalla, mientras que la línea 50 define una secuencia para desplazar el cursor una distancia fija a lo largo de la línea de fondo de la pantalla, situándolo en la columna correspondiente a cada bloque del histograma.

La línea 60 llama a una subrutina que representa la envolvente de los valores ADSR iniciales. Las líneas 70 y 80 establecen el volumen y la frecuencia del oscilador para la voz 1. Los valores de volumen pueden ir desde 0 (no se oye nada) hasta 15 (volumen máximo), por lo que puedes experimentar con ellos cambiando la segunda cifra de la línea 70. Las líneas 100 a 130 constituyen el bucle principal: el programa espera a que haya una pulsación de tecla, actualiza los valores de ADSR y presenta el histograma corregido.

La subrutina de la línea 500 garantiza que ha sido activada la envolvente 1 mientras se pulsa la tecla de espaciado. La subrutina de la línea 1000 actualiza los valores ADSR (líneas 1000 a 1080), POKEandolos en los registros adecuados (líneas 1090 a 1100). Las líneas 1200 a 1350 calculan el histograma, evaluando las cuatro fases A, D, S y R. Las variables XX y HT representan las coordenadas X y altura de cada columna.

La subrutina de la línea 200 representa la columna de HT en la posición XX, utilizando las posiciones anteriormente mencionadas. El cursor se coloca al fondo de la pantalla, en la columna XX, lo cual se hace en las líneas 2000 y 2010. La línea 2020 dibuja la columna.

CONFORMACION DE LA ONDA

Las fases ADSR para la voz 1 se controlan con los *nibbles* (medio byte) que hay en las posiciones 54277 y 54278; el valor de ataque está controlado por el *nibble* superior de la dirección 54277, la caída inicial por el *nibble* inferior de esa misma dirección, el nivel de sostenimiento por el *nibble* superior de 54278 y el abandono final por el correspondiente *nibble* inferior.

Para hacer audible un oscilador, hay que seleccionar una forma de onda y hay que disparar su envolvente conformadora; esto se hace poniendo a 1 los bits del registro que controla la voz (dirección 54276, línea 500 para la voz 1). El bit 0 de este registro es la «puerta» o «disparo» de la envolvente conformadora de esa onda. Su acción es parecida a la de la pulsación de una tecla en un órgano: mientras está pulsada, la nota sigue sonando. Cuando se suelta la tecla, la nota entra en su fase final de relajamiento.

Los registros de control también seleccionan y conmutan la forma de onda a utilizar: poniendo a 1 el bit 7 se selecciona ruido aleatorio, el bit 6 selecciona una onda pulsada, el bit 5 selecciona una onda en diente de sierra y el bit 4 una onda triangular. El bit que controla la envolvente ha de combinarse con el bit que controla la forma de onda. Por ello la instrucción POKE 54276,33 selecciona una onda en forma de diente de sierra y activa la puerta, mientras que POKE 54276,32 desactiva la puerta (poniendo a 0 el bit 0) pero mantiene la forma de onda elegida, iniciando así la fase final. La instrucción POKE 54276,0 desactivaría la puerta pero también dejaría de seleccionar la forma de onda, silenciando la voz.

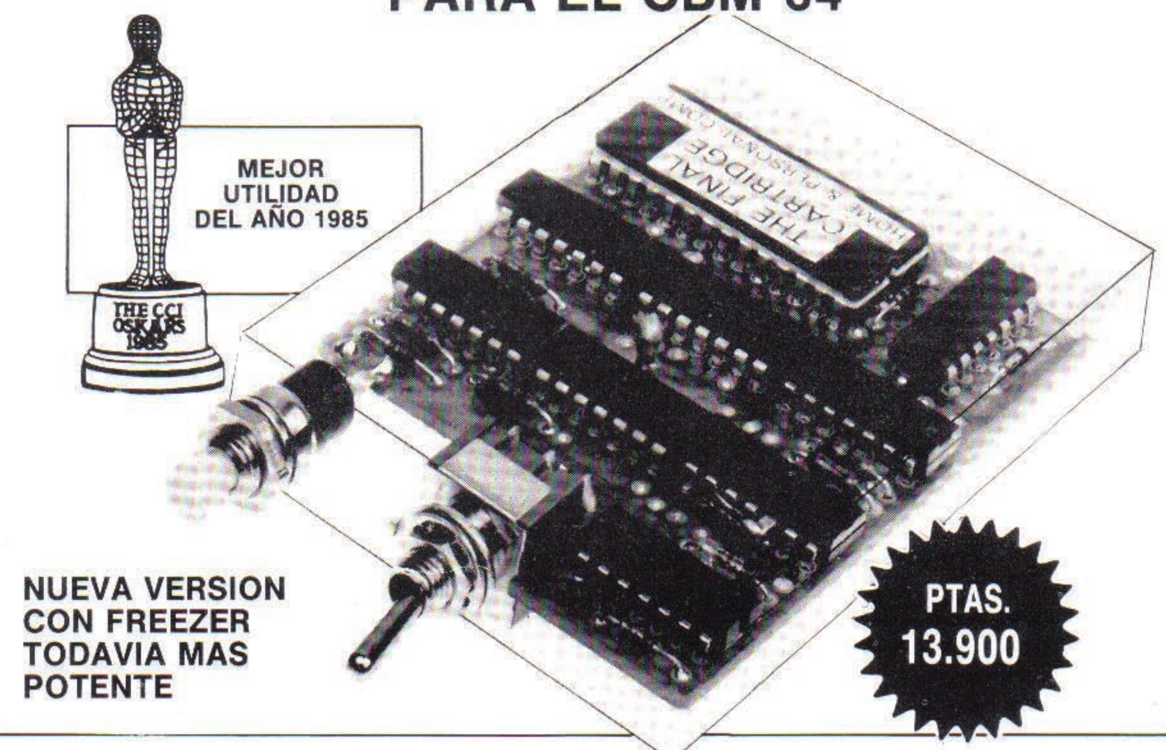
INTERDEPENDENCIA DE LAS FASES

Para hacer el mejor uso de la envolvente de ADSR, es importante entender cómo interactúan estas fases entre sí. Considera lo que sucedería si el valor de sostenimiento es 15, el máximo posible. El nivel máximo, alcanzado al final de la fase de ataque, será el mismo que el nivel de sostenimiento, por lo que no se produce una caída de uno a otro, y la envolvente resultante será igual independientemente del valor de caída.

Como segundo ejemplo, supone que el valor de sostenimiento es 0, la caída inicial es muy larga y el abandono final es también muy largo. Al final de la fase de ataque, la nota empieza a decaer hacia cero a la velocidad inicial de caída. Si se desactiva la puerta más bien pronto (pero antes de que se alcance el nivel de cero), paradójicamente la nota resultará prolongada, debido a que se ha entrado en la fase de caída más lenta.

THE FINAL CARTRIDGE

EL PRIMER SISTEMA OPERATIVO EXTERNO PARA EL CBM 64



Este nuevo sistema operativo construido en un cartucho no utiliza memoria alguna y está siempre presente. Compatible con 98% de los programas.

TURBO DISCO: Carga 6 veces más rápido - Salva 8 veces más rápido - No Borra la pantalla.

TURBO CINTA: 10 veces más rápido, incluso con ficheros. Utiliza los comandos normales del Commodore (Load-Save-Input#,...).

INTERFACE CENTRONICS: Compatible con las impresoras de tipo paralelo. Imprime los gráficos Commodore y los códigos de control. (Importante para los listados).

VOLCADOS DE PANTALLA: De alta resolución y texto. 1 página de ancho. 12 tonos de grises, copia pantallas de juegos o de programas como Doodle, Koala Pad, Print Shop, etc. Busca automáticamente la dirección en memoria.
Versión especial para impresoras MPS 801 y 803.

24 K. MAS DESDE EL BASIC: 2 nuevos comandos, "memory write" y "memory read" mueven 192 bytes muy rápidamente en cualquier sitio de los 64K RAM del CBM 64. Se pueden usar con cadenas y variables.

COMANDOS DEL BASIC 4.0: como dload, dsave, dappend, catalog, etc.

TECLAS DE FUNCION PREPROGRAMADAS: Run, load, save, catalog, comandos de disco, list (quita las protecciones contra el listado).



Ejemplo de volcado de pantalla.

OTRAS AYUDAS: Puede borrar parte de una línea, parar y continuar los listados, mover el cursor a la izquierda.

MONITOR DE CODIGO MAQUINA: Scroll hacia arriba y abajo. Bankswitching (para leer y escribir debajo de las ROMS), etc... No reside en memoria.

BOTON DE RESET: Resetea el monitor.

Resetea con volcados de pantalla.
Resetea todos los programas protegidos.

INTERRUPTOR ON/OFF: Por si acaso...

FREEZER:

Para y continúa casi todos los programas. Permite hacer una copia del programa a disco o a cinta automáticamente.

—copia todo el programa en 1 bloque.
—es 4 ó 6 veces más rápido que los productos (similares) y especializados que se pueden encontrar en el mercado americano.

Funcionamiento sencillo por menús.
Permite: —impresión página entera.
—cambio de color letras o fondo.
—modo normal o inverso.
—salto a monitor o reset.
También, modo entrenamiento para juegos (anula la detección de colisión de los sprites).

Disponible en las mejores tiendas o directamente por correo a:



ASI ES EL MAPA DE MEMORIA DEL COMMODORE 64 (II)

Continuamos analizando en este artículo las direcciones del 64, una a una.

Byte $000F = 15

Este byte resulta imprescindible durante la ejecución de la rutina LIST. Su contenido indica la existencia de comillas, visualizando directamente lo que va a continuación, en lugar de *destokenizarlo* como si se tratase de una palabra reservada del lenguaje BASIC. Es decir, lo trata como cadena de caracteres.

Igualmente se emplea como avisador de errores, tales como el agotamiento de la memoria disponible.

Igualmente almacena información intermedia durante el proceso de *tokenización* de una línea de programa recién tecleada.

Byte $0010 = 16

Esta dirección también actúa como un *flag* (indicador) de uso multiple. Interviene durante la rutina de com-

```
Byte No.     Contenido
========     =================

$0016......Apuntador hacia
           $19,$1C,$1F o $22
$0017......Valor 3 unidades
           menor que el de
           la direccion $22
$0018......Contiene #$00
-------------------------------

        PRIMER DESCRIPTOR
$0019......Longitud cadena
$001A......Byte bajo de la
           direccion de la
           cadena 1.
$001B......Byte alto del
           anterior
-------------------------------

        SEGUNDO DESCRIPTOR
$001C......Igual`que $0019
           para 2a. cadena
$001D......Igual que $001A
           para 2a. cadena
$001E......Igual que $001B
           para 2a. cadena
-------------------------------
$001F......Igual que $0019
           para 3a. cadena
$0020......Igual que $001A
           para 3a cadena
$0021......Igual que $001B
           para 3a. cadena
-------------------------------
```

```
---256*PEEK(56)+PEEK(55)---
    Final area del BASIC

   Zona para las cadenas
---256*PEEK(52)+PEEK(51)---
   Principio de cadenas

       Memoria libre
--256*PEEK(50)+PEEK(49)+1--
  Principio memoria libre

  Zona para las matrices
--256*PEEK(48)+PEEK(47)+1--
   Principio de matrices

   Zona de las variables
--256*PEEK(46)+PEEK(45)+1--
  Principio de variables

  Zona del programa BASIC
--256*PEEK(44)+PEEK(43)+1--
    Comienzo del BASIC
```

Grafico de la distribucion del mapa de memoria para el BASIC y el modo de conocer su particion en cada momento. Solo con imprimir(con PRINT) la operacion con PEEKs que nos interese. Excepto las de comienzo y final, todas son alteradas frecuentemente.



probación de si es aceptable un determinado nombre de matriz, que detecta incluso si se trata de una matriz o una función del tipo FN.

Byte $0011 = 17

La forma de actuar de las palabras clave del BASIC: INPUT, GET y READ es muy parecida y comparten diversas subrutinas del código máquina que las ejecuta. Este byte es el responsable de que entren en acción unas si y otras no, dependiendo de la palabra, y que los mensajes de error posibles sean los adecuados.

Con INPUT el valor es 0, 64 para GET y 152 para READ.

Byte $0012 = 18

Las rutinas de comparación, que emplean los signos =, utilizan este byte para indicar el resultado de la(s) misma(s).

También sirve para indicar el caracter positivo o negativo del resultado de TAN o SIN.

Byte $0013 = 19

Aquí es depositado el número de fichero lógico (CMD más algo) con el que se comunica el sistema.

Byte $0014 y $0015 = 20 y 21

Estos dos bytes contienen un número de 2 bytes de longitud, empleando el formato de byte bajo/byte alto.

Dicho número es el de la siguiente línea a que ha de dirigirse el programa cuando se encuentran las palabras GOTO, GOSUB, ON, LIST. También almacena las direcciones de la memoria a las que deben acceder SYS, PEEK, POKE o WAIT.

Byte $0016 = 22

El contenido de este byte apunta hacia el comienzo del siguiente lugar disponible en la pila (área de memoria reservada en este caso entre las direcciones $0019 a $0022) utilizable por hasta tres descriptores de otras tantas cadenas alfanuméricas temporales.

Cuando la pila ha sido llenada, apuntará hacia el byte $0022, que no es utilizable.

Bytes $0017 y $0018 = 23 y 24

Estos dos bytes son complementarios del anterior.

Empleando la fórmula byte bajo/byte alto, actúan como índice apuntador hacia la última dirección utilizada en la pila de descriptores de cadenas temporales.

Bytes $0019 a $0021 = 25 a 33

Esta es la aludida pila de descriptores para cadenas temporales, que contiene la información referente a cadenas alfanuméricas no asignadas.

Cada uno de los tres descriptores que caben aquí está compuesto por tres bytes, que contienen la siguiente información: Longitud de la cadena y bytes bajo/alto de la dirección de la memoria en que comienza.

Bytes $0022 a $0025 = 34 a 37

Esta es un área temporal de trabajo utilizada para diversos fines.

Byte 00$26 a $002A = 38 a 42

Area de trabajo empleada por las rutinas que realizan el producto y la división.

Bytes $002B y $002C = 43 y 44

Ambos bytes alto/bajo actúan como apuntador de comienzo del área de memoria que contiene el programa en BASIC. Normalmente contienen la información ($01) en $002B y ($08) en BASIC, lo que es igual a la dirección decimal 2049.

Bytes $002D y $002E = 45 y 46

Este es otro apuntador de dos bytes, que indica el comienzo del área de memoria en la que se almacenan las variables utilizadas por el programa BASIC. Es igualmente la dirección final del área empleada para almacenar el programa (la zona de memoria que nos interesa comienza una dirección más arriba).

Bytes $002F y $0030 = 47 y 48

Estamos ante otro apuntador, que define el final de área de almacenamiento de las variables que no sean tipo matriz e indica el comienzo del área para almacenar las matrices.

Bytes $0031 y $0032 = 49 y 50

Nuevamente, este otro puntero indica el final del área utilizada por las matrices. En las direcciones superiores queda un área libre de memoria utilizable, antes de llegar al área empleada por las cadenas alfanuméricas.

Bytes $0033 y $0034 = 51 y 52

Puntero indicador del final del área libre de memoria y el principio de la zona en que se almacenan las cadenas definidas en un programa BASIC.

Bytes $0035 y $0036 = 53 y 54

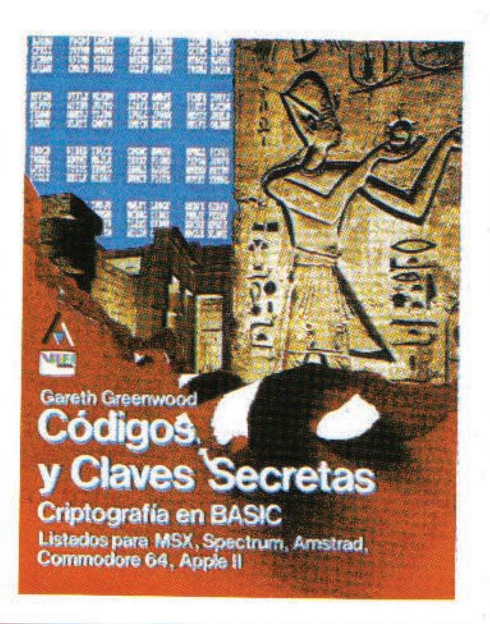
Este es un puntero de utilidad temporal para el manejo de las cadenas alfanuméricas.

Bytes $0037 y $0038 = 55 y 56

El contenido de estos dos bytes define la máxima dirección utilizable desde el BASIC y su alteración permite crear una zona de memoria «protegida».

CONCURSO INPUT/ANAYA MULTIMEDIA

¡DESCUBRE AL ASESINO!



BASES:

Durante tres números sucesivos irán apareciendo en **INPUT** las tres partes del relato «El caso del anciano asesinado» junto con los mensajes cifrados que constituyen las respuestas parciales a la solución del crimen. Cada mes, quienes logréis descifrarlos participaréis en un sorteo consistente en 10 lotes de libros por un valor de 15.000 ptas., a elegir del fondo editorial de **Anaya Multimedia**, y una suscripción gratuita por un año a **INPUT**. Habrá tres sorteos, de tal forma que no será necesario haber descifrado el enigma del mes anterior para optar al premio.

Un fabuloso premio será la guinda final de este pastel. El descubridor de las motivaciones del crimen visitará al próximo **PCW Show** a celebrar en Londres. En caso de haber más de un acertante, recurriremos a la inevitable fórmula de sorteo.

La solución del primer mensaje cifrado deberéis enviarla, junto con vuestros datos personales, a la Redacción de **INPUT** antes del 15 de junio.

Las decisiones del jurado serán inapelables, dándose en las páginas de **INPUT** cumplida cuenta de la marcha del concurso.

Nota: El libro de reciente aparición «Códigos y claves secretas», de ***Anaya,*** *contiene todos los programas útiles para esta labor. Aunque si has trabajado para los servicios de inteligencia de alguna potencia no te será de gran utilidad.*



EL CASO DEL ANCIANO ASESINADO

Repasé la casa y comprobé que el viejo no fumaba, aunque no carecía de algunos vicios cuya enumeración no aportaría nada a este informe.

—El ladrón o los ladrones debieron dejarse un cigarro encendido —comentó mi ayudante.

—Los ladrones no suelen fumar mientras trabajan —respondí.

Di un par de vueltas más, que sólo me sirvieron para comprobar la porquería que es capaz de acumular en una casa un hombre solitario, aun cuando sea tan rico como aparentaba nuestro cadáver maniatado. Revisé de nuevo el interior de la caja fuerte. Había cartas de amor, recortes de periódicos, facturas sin saldar, apremios de pago, títulos de propiedad, una dentadura postiza y una póliza de seguro. Me guardé la póliza y dejé lo demás donde estaba.

En esto, llegó el juez con intención de levantar el cadáver. Tuvo que esperar un poco, porque nuestros expertos no habían terminado de estudiar el modo en el que el viejo había sido maniatado. Al poco, apareció también por el piso un anciano muy parecido al muerto y que dijo ser hermano suyo.

—¿Cómo se ha enterado Vd. de todo esto? —pregunté.

—Vivo en el piso de arriba. Estaba durmiendo la siesta, pero me ha avisado un vecino —respondió llorando frente al espectáculo.

Entonces, mi ayudante me pasó una nota que decía:

```
XNLZNJSYJ HNKWFIT HTQZRSFX HTRUQJYFX XFI
KQ UISIN MFTSN QXZSJ FSFWS TFRSJ JXJYS Z
XFJV FISNF XJJIW JNTTS IXUTF NJRQW YQZTW
 SNTYN JQ
```

Desde la comisaría telefoneé a la casa de seguros de la póliza y me confirmaron lo que ya había leído: que cubría un seguro por robo de una serie de joyas valoradas en quince millones de pesetas, joyas que, según la información de la casa de seguros, permanecían en una caja fuerte de la casa del anciano muerto. Al día siguiente visité a su hermano y le sometí a un complicado interrogatorio, complicado porque el pobre viejo no paraba de llorar.

Juan José Millas

LOS MEJORES DE INPUT COMMODORE

PUESTO	TITULO	PORCENTAJE
1.°	Commando	27,1 %
2.°	Rambo	15,7 %
3.°	Hardball	12,8 %
4.°	Uridium	8,5 %
5.°	Green Beret	7,1 %
6.°	Pitstop II	7,1 %
7.°	Broad Street	5,7 %
8.°	Kung Fu Master	5,6 %
9.°	Saboteur	5,5 %
10.°	Time Tunnel	4,9 %
		100 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Junio de 1986

SOFTACTUALIDAD

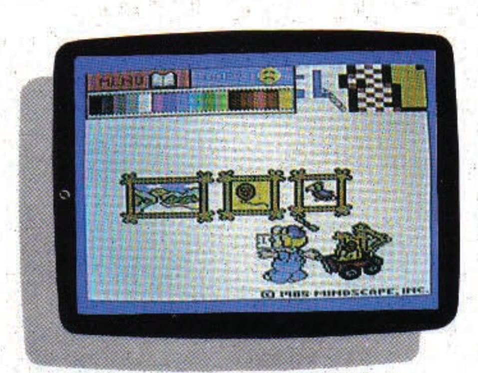
EL NUEVO GOLF DE ACCESS

Steve y **Bruce Carver**, los creadores del **Beach Head** y **Beach Head II**, se preocuparon por lanzar en el *show* de Las Vegas, celebrado en enero, un nuevo programa de excelente nivel: el **Leaderboard**, se trata de un excelente *golf* que barre a todos sus predecesores. Esto no es una casualidad, las características del programa son muy especiales: 78 hoyos, visión tridimensional del campo, opción de *Snap*, efectos de tiro, 3 niveles de juego, 4 pistas diferentes, todo ello con una calidad gráfica soberbia, en la que los movimientos del jugador parecen una película animada. Estamos ante uno de los mejores juegos del año.



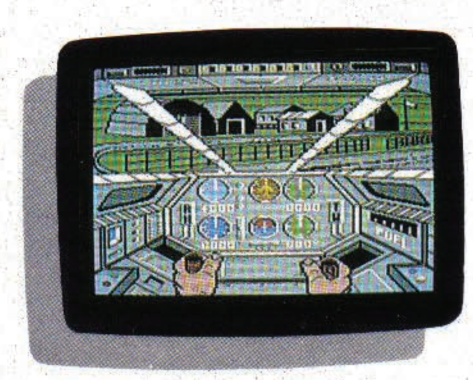
SHAPE ME:

Estamos ante un nuevo programa de dibujo y quizás sea éste el mejor ya que es capaz de trabajar sin desteñir los colores. Es de la casa **Sprout** y en principio parece que es un programa orientado a los niños, las opciones de relleno son innumerables, la selección desde el menú es muy fácil y asequible y la ristra de impresora con las que se puede trabajar es innumerable.
La novedad quizás más relevante del programa es la del *Zoom*, que trabaja como una ventana, ocupando la cuarta parte de la pantalla total (al igual que **Gammes Maker** que os comentamos previamente en esta sección) y que permite realizar todas las opciones (círculo, caja, línea, aerógrafo, etc...) permaneciendo en este modo.



BIPENETRATOR

Bipenetrator es un fenomenal juego de simulación de vuelo y estrategia conjuntamente. Has de rescatar ciertos documentos situados en un campo enemigo fortificado. El objetivo que has de alcanzar está situado muy lejos y se halla muy protegido, para cumplir la peligrosa misión deberás utilizar un helicóptero cargado de explosivos ya que tus enemigos no dudan en hacer todo lo posible para aniquilarte.
Los gráficos sorprenden tanto como



los de **Accolade**, el sonido de los misiles tampoco se queda manco. Todo ello hace que incluso para los no «amateurs» de juegos de simulación de vuelo sea un programa muy atractivo.

ELECTRA GLIDE

La nueva compañía inglesa **English Software**, se ha decidido por fin a sacar a la venta el muy esperado **Electra Glide**, juego de carreras de coches, con la esporádica aparición de obstáculos en la carrera que te



van obligando a hacer *slalom*.
Hay un *scrool* en tres sentidos diferentes (recordad el **Summer Games II** y su prueba de caballo), opción a coger tres tipos diferentes de coches (según pruebas de termodinámica), tres situaciones diferentes (Australia, Gran Bretaña, y Estados Unidos).
La conducción del coche es rápida atravesando tuneles, zonas rocosas, bordeando lagos, etc... La impresión de velocidad está más lograda que en otros programas.

BOMB JACK

El popular juego de las **Bombas de Jack** ha llegado por fin a nuestro micro. Realmente es un juego simplote, aunque enseguida te picas y sientes deseos de llegar más lejos y de pasar más y más pantallas. La primera pantalla, quizás la más conocida de todas, es la que tiene por escenario el **Fenix** egipcio. Lo que has de hacer es muy fácil (aparentemente): quitar las numerosas bombas que llenan la pantalla. Hay que mencionar que este juego ha tenido una acogida excepcional en Inglaterra debido a las cuotas de popularidad que había alcanzado la versión del programa en los recreativos de los bares.

VISITANTES!

ZONA 5

HANGAR

ZONA 4

REACTOR NUCLEAR

ORDENADOR CENTRAL

ZONA 3

PURIF. DEL AIRE

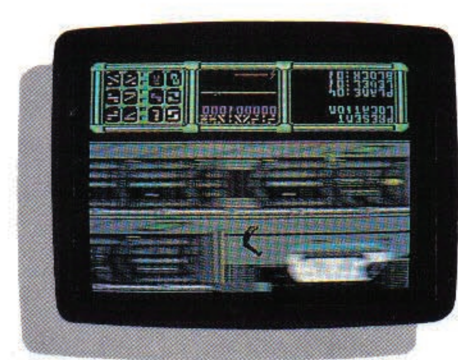
ENTRADA DEL AGUA

ZONA 2

ZONA 1

LOS VISITANTES ACECHAN DE NUEVO





Los repulsivos lagartos con piel de cordero (leáse humana), que angustiaron al planeta con sus sueños de gloria en la serie televisiva **V**, han llegado desde el confín del universo para adueñarse de la memoria de tu **Commodore 64**.
Tu eres el inefable **Donovan** (destacado lider de los comandos de liberación terrícola).
El juego comienza en el instante en que logras infiltrarte en la nave estrella de los temibles alienígenas.
El propósito que persigues es evidente: minar con cargas explosivas el interior de la nave nodriza. En tu contra tienes a la malvada **Diana** (tenía que ser ella obligadamente), comandante en jefe de los visitantes. Sus órdenes de seguridad para los sofisticados robots que custodian el platillo son terminantes. Los robots están aleccionados para detener a nuestro aventurado protagonista a cualquier precio y para ello existen cuatro modelos altamente especializados: los llamados de mantenimiento, que viajan por la superficie de los pasillos; los robots de limpieza, que se desplazan a la altura de la cabeza y los mucho más peligrosos de vigilancia y seguridad. Los primeros informan al ordenador central de la nave de cualquier paso en falso o movimiento sospechoso. Los segundos disparan a matar.
Todo sistema artificial tiene un punto de peligro imprevisto y estos robots se alimentan de energía de alta tensión, por lo que cualquier choque o disparo procedente de ellos supondrá una terrrible descarga eléctrica en las carnes de **Donovan**.
En la parte inferior de la pantalla aparece el electrocardiograma de **Donovan** y si llega a ser una línea plana, significará que nuestro héroe se fue a un mundo mejor (esperemos que sin lagartos!).
Para tener movilidad por las estancias y plantas de la nave, has de conseguir ubicarte encima de alguno de los tapices, que verás esparcidos por el suelo, y dirigir tu *joystick* hacia abajo.
Los gráficos son bastante agradables, aunque algo sobrecargados, haciendo que la imagen sea confusa. El movimiento del *sprite* se parece curiosamente al de **Misión Imposible**, pero no está tan logrado. Tal vez se echa de menos un mayor surtido de opciones de acción en el juego, algo más que sólo disparar.
No obstante no es un juego fácil de dominar en primera instancia, sobre todo si no conoces aún el alfabeto marciano. Ello te permite acceder al ordenador central a través de los terminales y abrir las puertas o manejar el láser. Un consejo, busca en los laboratorios el letal polvo rojo.

DATOS GENERALES

TITULO V

FABRICANTE Ocean

ORDENADOR Commodore 64

TEMA DEL PROGRAMA
Invasion marcianos

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**8**
INTERES	**8**
GRAFICOS	**9**
COLOR	**9**
SONIDO	**-**
TOTAL	**34**

****

EL PROYECTO HALLEY

Desde hace algún tiempo, algunos científicos creen que la nube o cola del cometa **Halley** es capaz de mantener vida; no vida como podriamos entenderla a primera vista, sino vida en cuanto a existencia de bacterias.
Otros científicos no piensan que esta teoría pase de ser fruto de una imaginación calenturienta y se pasan

DATOS GENERALES

TITULO The Comet Game

FABRICANTE Firebird

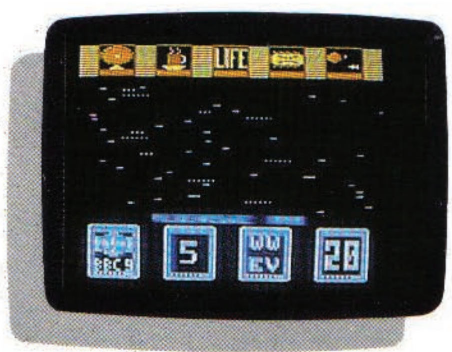
ORDENADOR Commodore 64

TEMA DEL PROGRAMA
En torno al Halley

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**9**
INTERES	**8,6**
GRAFICOS	**8,8**
COLOR	**8,5**
SONIDO	**8**
TOTAL	**42,9**







la vida riendo cada vez que algún despistado menciona esta extravagancia: ¿Cómo un enorme trozo de materia que recorre el frío sideral a más de 200.000 Km/hora puede soportar vida alguna?
Esta es la base del juego y el papel que has de desempeñar es el de recorrer el camino seguido por el cometa hasta alcanzarlo, resolviendo con gran prontitud todos aquellos problemas que te vayan surgiendo. Finalmente, cuando alcances el cometa, habrás de destruir los gérmenes que lo pueblan ya que estos son dañiños para la vida humana.
En el camino que habrás de recorrer hasta alcanzar la nave, te encontrarás con guardias de seguridad interplanetaria que no dudan en lanzarte salvas de misiles. Tu defensa para detener estos mortíferos visitantes consiste en lanzar botes de plasma. Juego distinto a muchos otros, con gráficos excelentes y mucha emoción.

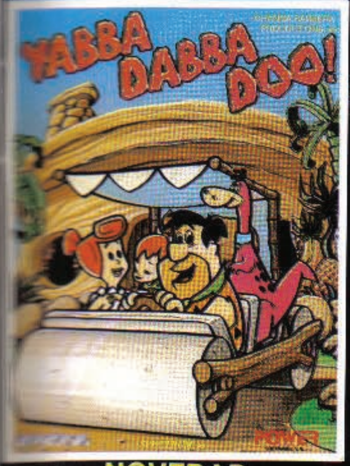
POWER

SOFTWARE, S.A.

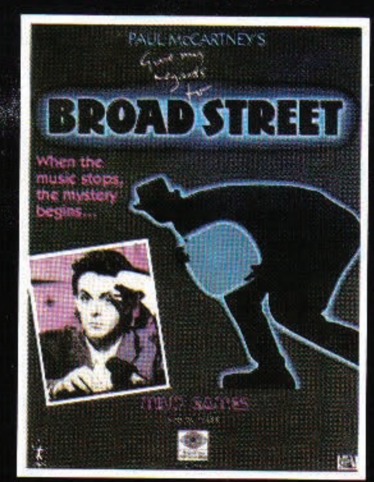
NAPOLES, 98, 1.° 3.ª - Tels. 232 24 61 - 232 25 52

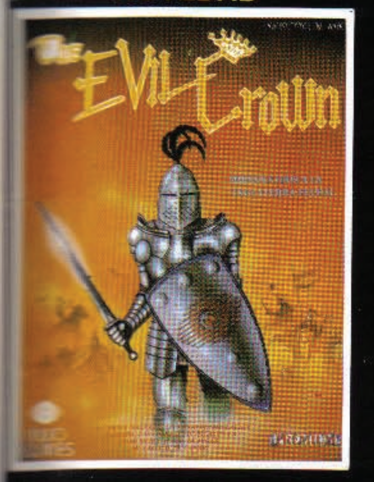
08013 BARCELONA (SPAIN)

JUEGOS INTELIGENTES PARA MENTES INTELIGENTES



NOVEDAD







SPECTRUM

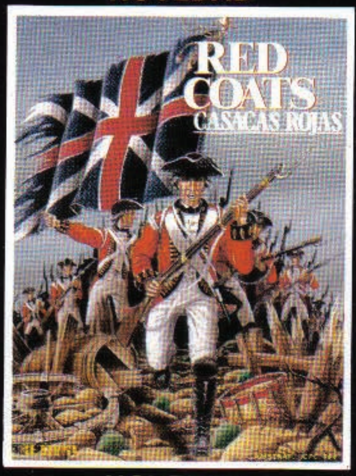
BROAD STREET	2.400 P.V.P.
EVIL CROWN	2.400 P.V.P.
TREASURE ISLAND	2.400 P.V.P.
YABBA DABBA DOO	1.800 P.V.P.
RED COATS	1.800 P.V.P.
SCHIZOFRENIA	2.400 P.V.P.
OVERLORD	1.800 P.V.P.
ANT-ATTACK	1.700 P.V.P.
FALL OF ROME	1.800 P.V.P.
GATECRASHER	1.400 P.V.P.



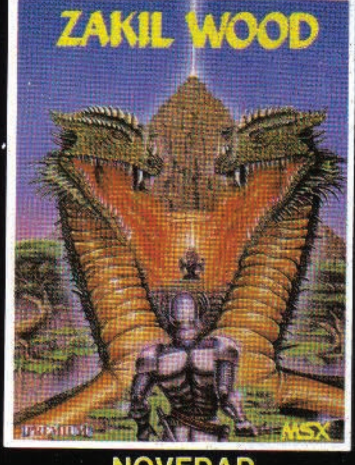
NOVEDAD



NOVEDAD



NOVEDAD



NOVEDAD

COMMODORE

BROAD STREET	Cas. 2.600 – Disc: 3.600 P.V.P.
EVIL CROWN	Cas. 2.600 – Disc: 3.600 P.V.P.
TREASURE ISLAND	Cas. 2.600 – Disc: 3.600 P.V.P.
YABBA BADDA DOO	Cas. 1.800 – Disc: 3.100 P.V.P.
SCHIZOFRENIA	Cas. 2.600 – Disc: 3.600 P.V.P.
ANT ATTACK	Cas. 1.700 – Disc: 2.800 P.V.P.
FALL OF ROME	Cas. 1.800 – Disc: 2.800 P.V.P.
SEE SAW	Cas. 1.600 P.V.P.
TRASHMAN	Cas. 1.600 P.V.P.

AMSTRAD

YABBA DABBA DOO	Cas. 1.800 – Disc. 3.100 P.V.P.
RED COATS	Cas. 2.100 – Disc. 3.100 P.V.P.

M.S.X.

ZAKIL WOOD	Cas. 2.800 P.V.P.
GAMES DESIGNER	Cas. 2.400 P.V.P.

Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos

Nombre ____________________

Dirección ____________________

Teléfono ____________ Ciudad ____________

Firma:

Título ____________ Sistema ____________ Precio ____________

Título ____________ Sistema ____________ Precio ____________

Título ____________ Sistema ____________ Precio ____________

Título ____________ Sistema ____________ Precio ____________

☐ Contrarrembolso. ☐ Adjunto Talón. ☐ Giro Postal.

Deseo recibir información de sus programas en: MSX ☐ MASTRAD ☐

EL EIDOLON





DATOS GENERALES

TITULO Eidolon

FABRICANTE Lucasfilm

ORDENADOR Commodore 64

TEMA DEL PROGRAMA Laberinto fantástico

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**8**
INTERES	**8**
GRAFICOS	**9**
COLOR	**7**
SONIDO	**8,5**
TOTAL	**40,5**

El Eidolon utiliza la misma técnica para los paisajes de fondo que los programas **Ballblazer** y **Koronis Rift**, sólo que en este caso la acción tiene lugar bajo tierra. La técnica aludida es la paisajes fractales (los lectores habituales de INPUT ya sabéis de que hablamos).

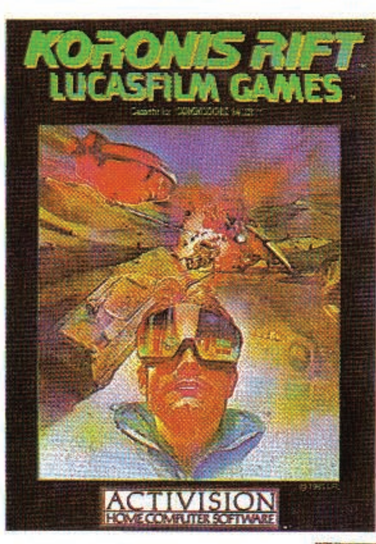
El Eidolon es una pequeña nave que has de conducir a través de un laberinto de cuevas. Buscas fuentes de energía y tesoros, mientras te deshaces de algunas criaturas que parecen salidas de una pesadilla. Existen diferentes tipos de materiales energéticos, que habrás de recoger o evitar. También tienes que capturar una colección de cristales de diferentes colores, que te permiten el adentramiento en las profundidades. Al final de cada nivel de progresiva dificultad, encuentras un dragón malvado, que debes aniquilar.

La experiencia acumulada dirá la última palabra.

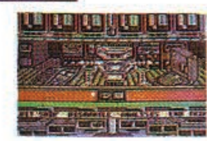
El estilo que **Lucasfilm** ha imbuido en sus juegos está claramente visible en este programa.

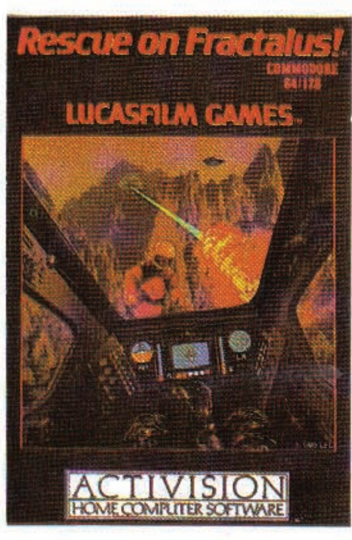
Tanto los gráficos como el sonido son superiores.

Lo mejor de LUCASFILM GAMES™ para tu commodore 64/128



Encuentra un tesoro en un remoto sistema estelar, pero ¡cuidado! con sus guardianes. Si quieres sobrevivir elabora un cuidadoso plan. La recompensa esta fuera de este mundo.

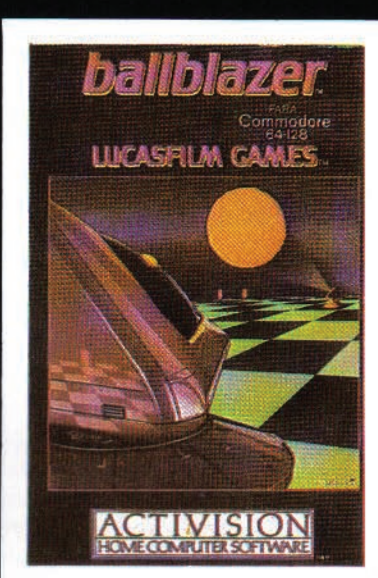




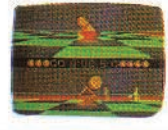
Recorre un planeta en tres dimensiones, a la búsqueda de tus compañeros, mientras el enemigo te persigue.

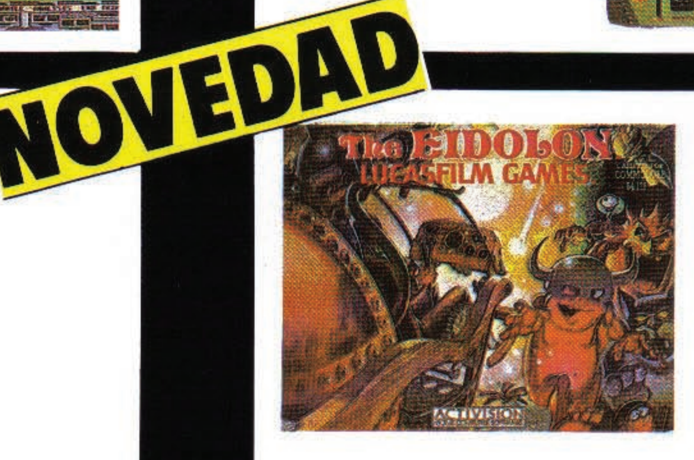


NOVEDAD



Juega en la ronda final del campeonato interestelar de Ball Blazer y consigues el honor más elevado para tu planteа: El Master Blazer.





Descubre los secretos de Eidolon a través de una misteriosa máquina del siglo XIX, y sumergete en un mundo subterráneo poblado de curiosas criaturas máginas y feroces dragones guardianes.





EN TIENDAS
ESPECIALIZADAS Y GRANDES ALMACENES,
O DIRECTAMENTE POR CORREO O TELEFONO A:

DISTRIBUIDO POR:
PROEIN, S.A.
Velázquez, 10 - 28001 Madrid   Tels. 91/276 22 08/09

TERROR EN VIERNES 13

Los juegos con tema de película encuentran continuidad en este programa.
El argumento coincide con el seguido en el propular *film*, en el cual la acción se desarrolla en un tranquilo campamento de verano.
Jasón, un audaz excursionista, muere ahogado. Su desconsolada madre jura entonces venganza en un intento por enjugar su tristeza. A la buena mujer no se le ocurre mejor idea que matar a todos los inocentes angelitos, excepto a una despierta chiquilla que logra ser más rápida.
Terror y espanto. **Jasón** se reencarna para acabar con los pocos que su madre no ha liquidado, por lo que se ve la familia es de aupa.
Empiezas el juego en el momento en que se supone que **Jasón** está cumpliendo con el curioso fin de ir mengüando la población juvenil en torno al lago.
Has de encontrar un lugar seguro donde no pueda entrar el maligno espectro y convencer a los pocos supervivientes de que desaparezcan sin dilacción de allí.
Lógicamente, como cualquier extraño que llega a un lugar, no conoces a nadie y puede que por error trates de convencer a **Jasón** de que se ponga a cubierto.
Por si fuera poco, **Jasón** puede ir poniendo en práctica su misión aniquiladora sin que tú estés presente. La acción criminal no se circunscribe a tí, sino que se puede desarrollar fuera de la pantalla que muestra el monitor. Cabe en este punto aclarar que existen multitud de diferentes ubicaciones, desde el cementerio hasta un pajar, pasando por diferentes habitaciones e incluso la iglesia. Los personajes deambulan a sus aires por todas partes sin reparar en tu presencia, aunque los persigas.
Cada vez que alguien cae se escucha un grito estremecedor y la representación gráfica del personaje, que hay en la base de la pantalla, se transforma en una tumba sin epitafio. También aparece el arma que eliges entre las que vas encontrando por el camino. Existe un total de diez diferentes, pero sólo puedes llevar una simultáneamente.
Como pistas podemos deciros que el lugar quizás más seguro sea la cruz del santuario. El modo de convencer a las pobres víctimas consiste en colocarse encima de ellos y, automáticamente, se dirigiran al lugar escogido. Lamentablemente estos incrédulos acompañántes se aburren dentro del escondrijo y continuamente intentan escaparse a dar una vuelta.
Un detalle simpático y espeluznante que acompaña a la cinta es una bolsita que contiene dos cápsulas de sangre sintética (claro) y comestible, que te hará disfrutar con mayor realismo de las matanzas de este «sangriento» programa.

DATOS GENERALES

TITULO Friday 13th

FABRICANTE Dommmark

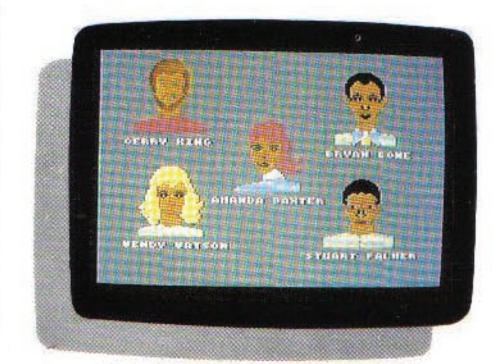
ORDENADOR Commodore 64

TEMA DEL PROGRAMA
Terror de película

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	8,5
GRAFICOS	7,8
COLOR	7,5
SONIDO	9
TOTAL	40,8



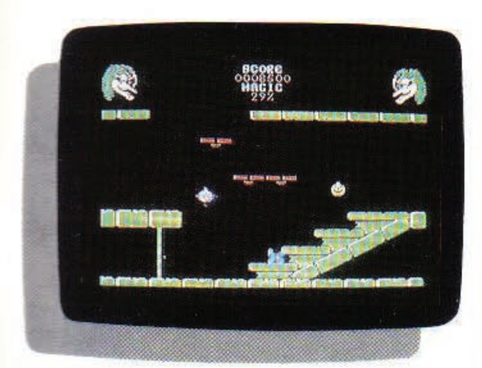


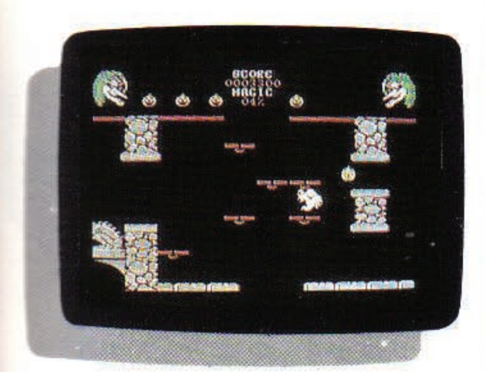






LAS BRUJAS DEL LUGAR







DATOS GENERALES

TITULO Cauldron II

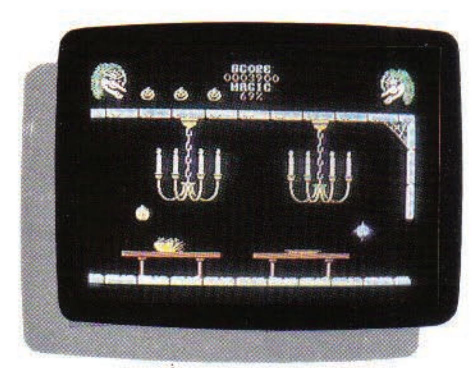
FABRICANTE Palace Software

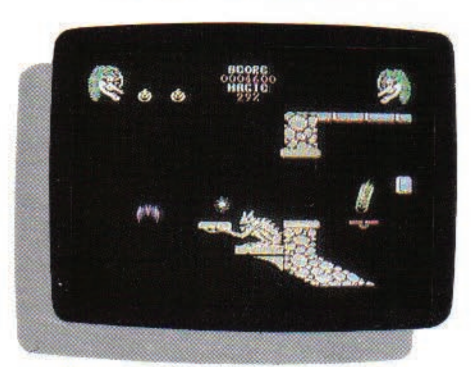
ORDENADOR Commodore 64

TEMA DEL PROGRAMA
Brujería

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	9
INTERES	8,6
GRAFICOS	9
COLOR	9
SONIDO	8,5
TOTAL	44,1





Esta es la continuación del popular **Cauldron** que lanzase hace tiempo la casa **Palace Software**.
Mientras se carga el juego, uno siente una curiosísima sensación: ¿no se habrán equivocado de título? Efectivamente, no existe gran parecido con el primer programa, incluso no aparecen los *sprites* del título original y existen grandes diferencias en los contenidos gráficos. El sonido es bueno, agradable y con cambios de melodía.
El juego es una acertada variante del típico de salas multiples, aunque sin *scroolling*. El *sprite* de la bruja se ha convertido, quizás por arte de magia, en una calabaza con ojos, nariz y boca que bota y bota sin cesar, efecto éste de gran realismo.
Tu calabaza saltarina ha de encontrar una serie de objetos dispersados entre las ciento ventiocho pantallas para romper un conjuro. Son siete en total, la tarea se ve dificultada por la aparición de espectros y demás monstruos, que dificultan el recorrido por la mansión y que desposeen progresivamente de sus poderes mágicos. El hecho de que mengüen estas fuerzas de lo invisible acaba con las vidas de que dispone nuestro protagonista.
Hay que ser cuidadoso con determinadas piedras, que al bascular te conducen a un oscuro pozo de duro fondo.
Los autores del juego justifican la mayor simplicidad de esta nueva parte alegando que no conocen a nadie que fuera capaz de llegar al final de la primera parte.

****

ROJOS CONTRA AZULES

Zoids es un juego adictivo y con un gran argumento digno de **Asimov**. **Zoids Star** es una estrella habitada por organismos vivientes que temen por su seguridad y dedican largos años a la investigacion de técnicas avanzadas de guerra.
Cuando acaban con el proceso de desarrollo, deciden crear Androides para las fortalezas volantes que han diseñado. Andorides bautizados como **Zoids** y que son destinados a dos tareas: Luchar entre sí como gladiadores y patrullar por los alrededores del planeta, previniendo posibles invasiones sorpresa.
Lamentablemente, un meteorito de dimensiones gigantes viene a perturbar esta paz, haciendo desaparecer a los organismos vivientes y dejando el planeta en manos de los Androides.
Antes de morir uno de los organismos vivientes lanza un SOS, que es recogido por una de las naves patrulleras. En la excitación del momento, un mal cálculo de

trayectoria hace que esta nave de salvamento se estrelle en la Luna azul, único satélite de **Zoids Star**. La diferencia de temperaturas y radiaciones de la Luna azul hace que los supervivientes del accidente tengan que volver a diseñarse ellos mismos adquiriendo un color rojo. Del día a la noche estos **Zoids** rojos, capitaneados por **Rojo el Terrible** se vuelven malos, muy muy malos.



Deciden unificar fuerzas, conquistar **Zoids Star** convirtiendo a los otros Androides (de color azul) en esclavos.
Ante la amenaza, los azules se agrupan y construyen un nave insigna llamada **Zoidszilla**, que hace que las fuerzas sean más equitativas entre ambos bandos. En ese momento una pequeña nave espacial se estrella en **Zoids** y el único





DATOS GENERALES

TITULO Zoids

FABRICANTE Hartech

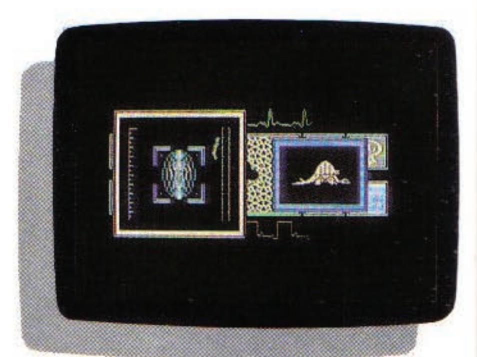
ORDENADOR Commodore 64

TEMA DEL PROGRAMA
Pelea entre androides

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**9**
INTERES	**8,8**
GRAFICOS	**9**
COLOR	**8,5**
SONIDO	**7,8**
TOTAL	**43,1**

superviviente humano se inserta en la sociedad de los **Zoids** azules y les ayuda a trazar un plan: Actuar como espía, entrar en la ciudad de los **Zoids** rojos y dejar allí un veneno que acabe con ellos, lamentablemente un misil lanzado por el propio **Rojo el Terrible** acaba con la nave **Zoidszilla** y sus tripulantes (el hombre incluido). Las seis piezas principales de la nave son



recogidas por **Rojo el Terrible** y puestas en las seis puertas de su ciudad.
El ánimo de los **Zoids** azules está en controlar sólo la inesperada aparición de otro hombre y su empeño por recuperar las seis piezas de la nave, las únicas que devolverán a los **Zoids** azules el valor y el potencial de guerra pueden salvar al planeta de la colonización. Este hombre eres tú.

★★★★★★★★★★★★★★★★★★★★★★★★★★★★★

EL ZOCO



Intercambio programas para C-64. Cinta o disco. También de Radio-Aficionados.EA3 FDX.

Eugenio Terrón.
Perez Galdos, 29, 2°.1ª.
Tel. 399 05 31.
Badalona (Barcelona).

Quisiera intercambiar ideas con los usuarios del C-64 o Vic 20.

Oscar Coes.
Tel. (93) 338 54 89.

Cambio juegos de última novedad por lápiz óptico para el C-64.

Luis José Monasterio.
Paco Pierra, 3, pta. 5.
Tel. 334 46 87.
46013 Valencia.

Intercambio programas para Commodore-64 y en disco unidad VIC. 1541. Así mismo estoy interesado en ponerme en contacto con chicos para formar un club en Madrid.

Carlos de la Presa Gomez.
Montepinar, 8.
Tel. (91) 854 00 33.
Guadarrama (Madrid).

Vendo cartucho SOCCER por 3.000 ptas. y cintas: Gremlins, Solo Flight, Bruce Lee, Raid Over Moscow, Pyjamarama y One on One, por 800 c/u. Todas 4.000 ptas. También los cambiaría por otros juegos.

Eduardo Pita Quintero.
Postigo 28, 2°.
Pontedeume (La Coruña).

Vendo juegos Atari con 6 cartuchos por 15.000 ptas.

Iñigo de Pineda.
Avenida Sarriá, 38.
Tel. 321 93 48.
Barcelona 08029.

Desearía comprar o intercambiar programas de todo tipo en cinta.

Miguel Angel Ortega.
Rodes, 26, 1° 1ª.
Tel. 338 09 77.
Hospitalet (Barcelona).

Compro o intercambio programas en cinta para CEM-64.

Javier Parpal Torrente.
Avda. de la Constitución 14, 4°.
Villena (Alicante).

Regalo MPS801 por 25.000 ptas. (con garantía) y C-64 por 25.000. Todo por 40.000 ptas. Todo en perfecto estado.

Juan Rafael O.M. Miledic.
Guayadeque, 4, 3° izq.
35009 Las Palmas de Gran Canaria.

Vendo o intercambio juegos originales del CBM 64. Abstenerse los intercambiadores de copias piratas.

Xavier Sancho.
Muntaner, 153 Pral. 4°.
08036 Barcelona.

Vendo Videopac Computer Philips G-7000 en perfecto estado con 2 joysticks y 3 cartuchos juegos por 15.000 ptas. (convertible en ordenador) o lo cambiaría por ZX Spectrum 48K.

Juan Carlos Chicón Dominguez.
Fco. Robert Graupera, 59.
Deltebre (La Cava) (Tarragona).

Intercambio de programas de interes y aplicación exclusivamente para el C-64, poseo cassette.

Jose Luis Torrebejano Estevez.
Gran Via Parque, 12, 1° F.
Tel. 23 50 86.
14005 Córdoba.

Vendo juego para Commodore 64 «The dambusters», que me han regalado por duplicado, por sólo 2.000 ptas.

Jaime Balsells.
Mariano Cuyner, 20.
Tel. (93) 803 39 64.
Igualada (Barcelona).

Cambio programas para C-64 en disco o cassette.

Miguel A. Ramos.
Avda. de Valladolid, 31, 6° B.
Tel. (988) 72 51 68.
34004 Palencia.

Intercambio programas en cinta para el C-64.

Cristobal Dávila.
Odena, 36, 3°.
Igualada (Barcelona).

Intercambio programas en cinta.

Jorge Boadilla Pérez.
Eduardo Dato, 8, 2° E.
Tel. (988) 74 95 39.
34005 Palencia.

Compro el datacassette original de la casa commodore. El cassete deberá estar en buen estado y sólo se aceptará la mejor oferta.

Eduardo Jimenez Abad.
Teniente Benitez, 7.
16003 Cuenca.

Quisiera vender o cambiar por cintas de juegos para el Vic 20, 4 cintas de programas para el Commodore 64. Precio a convenir. Me gustaría contactar con gente de Zaragoza que posean el Vic 20. !Responded pronto!

Javier Alfranco Gomez.
Teniente Ortiz de Zarate, 1, 3° izda.
Tel. (976) 51 62 09.
50015 Zaragoza.

Vendo Commodore 16 con unidad de cinta (datacassette 1531) nuevo; cassettes con juegos y libros. Todo por 27.000 ptas. Interesados escribir a:

José V. Alberte Tormo.
Luis Braille, 31, 3° D.
03010 Alicante.

Querría hacer un club de Commodorianos de Madrid, principalmente para cambiar juegos y programas. Escribir a:

Calle Brújula, 2, 7° A, escalera izda.
Tel. 676 30 31.
Torrejón de Ardoz (Madrid).

Cambio cartucho juego «Star Post» por otro cartucho. Cambio cinta 8 juegos con turbo, por otra idem.

Guillermo.
Teléf. (911) 22 10 29.

Desearía intercambiar programas (utilidades y juegos) para el Commodore 64.

Angel Pérez Martínez.
Tomás Bretón, 23 - 1°, 2°.
Teléf.(93) 372 92 94.
Esplugues de Llobregat (Barcelona).

Estoy interesado en cambiar todo tipo de programas para C-64.

José Lázaro Pérez.
Portugal, 39-41 -3°, 1ª.
08027 Barcelona.

Deseo cambiar juego del Commodore 64 por otro. Decathlon. Próximamente tendré más juegos.

Angel Sánchez.
Teléf. 619 50 56.
(De 13.30 a 14.30 h).
Madrid.

Cambio todo tipo de programas para C-64.

Alberto Andrés Torres.
Paseo de la Julia, 11.
34004 Palencia.



PROGRAMAS PROFESIONALES
Commodore 64 - 128

LIDER EN VENTA DE PROGRAMAS PROFESIONALES

CONTABILIDAD-128

SEINFO lanza al mercado el mejor programa de CONTABILIDAD de nuestros días.

CUENTAS

— Hasta 1.500 cuentas contables (programable).
— Número de niveles programable (hasta 4).
— Altas, bajas, modificación de cuentas.

APUNTES

— Contrapartida automática opcional.
— Control de cuadre de apuntes.
— Posibilidad de recomponer apuntes.

DIARIO

— Conceptos anuales o automáticos programables.
— Posibilidad de guardar diarios pendientes.
— Acceso a creación de cuentas desde diario.

EXTRACTOS

— Por pantalla o impresora.
— A partir de cualquier apunte.
— Por grupos de cuentas (desde/hasta).

INFORMES:

BALANCES

— De comprobación de sumas y saldos a cualquier nivel.
— Con criterios de selección de cuentas y períodos (desde/hasta).
— De situación (programable).
— Financiero por grupos de cuentas a determinar.

CUENTA DE EXPLOTACION

— Programable por el usuario.
— En cualquier fase del ejercicio.

LISTADOS CONTROL DE IVA

— Se genera a partir del diario de apuntes.
— Listado de IVA soportado y repercutido.

OTRAS CARACTERISTICAS

— 80 columnas con cualquier monitor.
— Un único disco de trabajo.
— Adaptado a cualquier impresora.
— Listados personalizados.
— Gestiona varias contabilidades.
— Códigos de cuentas de hasta 8 dígitos.
— Etc.

RENTAS-85

Programa para la declaración y autoliquidación del impuesto sobre la renta de las personas físicas del año 1985.

CARACTERISTICAS

— Toma de datos y cálculo.
— Posibilidad de grabar los datos.
— Salida por impresora opcional.
— Posibilidad de consulta en cualquier momento.
— Modificación y recálculo automático de la cuota.
(Disco o cassette)

GESTION COMERCIAL 25.000

PAQUETE INTEGRADO DE FACTURACION Y CONTROL DE STOCKS

Capacidades de ficheros programables por el usuario. (2.000 artículos, 1.000 clientes, 340 proveedores). Mantenimiento de ficheros. Entradas/salidas de almacén. Inventario permanente. Actualización automática de almacén. Reserva de pedidos. Distintos tipos de facturación. Control de I.V.A. Emisión de recibos. Diario de facturas. Conexión con CONTABILIDAD. Listados programables sobre cualquier fichero. Diseño a voluntad de la cabecera. Cálculo entre campos. Criterios de selección a voluntad (por provincia, fechas, ventas, compras, mínimos, zonas, familias, etiquetas, etc.).

CONTABILIDAD 25.000

Basada en el Plan Contable Español. 300 ó 1.000 cuentas. Contrapartida automática. Estractos por pantalla o impresora. Balances programables. Grupos 0 y 9. Balance de situación y cuenta de explotación programables.

ESTRUCTURAS 25.000

Calcula pórticos planos de hormigón armado. Calcula los esfuerzos para las tres hipótesis verticales, viento y sismo. Amado total de vigas y pilares. Cuadro de pesos de hierro. Cuadro cúbico de hormigón. Listado de todos los esfuerzos en el armado.

MEDICIONES Y PRESUPUESTOS 25.000

Programa de mediciones y presupuestos de obra totalmente programable por el usuario. Listado de mediciones y presupuesto por partidas. Posibilidad de ajuste automático de presupuesto.

FACTURACION 15.000

Programa de facturación directa. Fichero de artículos y clientes. Diarios de ventas. Desglose de impuestos. Emisión de recibos. Varias versiones.

CONTROL DE STOCKS 15.000

Ficheros de artículos y proveedores. Control de entrada/salida de almacén. Actualización automática. Inventario permanente. Inventario bajo mínimos. Listados varios.

TRATAMIENTO DE TEXTOS 5.850

Teclado castellano. Fácil manejo.

(PIDA INFORMACION SOBRE OTROS PROGRAMAS PROFESIONALES Y TECNICOS)

Pida información **SEINFO, S. L.** Avda. Goya, 8 - 50006 ZARAGOZA (976) 22 69 74-23 29 61



MANTENIMIENTO Y REPARACION DEL FLOPPY 1541

Autor: Herrmann
Editor: Data Becker (Ferre Moret)
Páginas: 220
Precio: 2.800 ptas.

Se trata desde luego de una obra curiosa, nada menos que una serie de consejos, técnicas, procedimientos, esquemas y listas de componentes con las que acometer un extraño tipo de *bricolage*: el mantenimiento y la reparación casera de la unidad de *diskettes* **1541** de **Commodore**. La idea es buena y la realización del libro también. Empieza éste dando algunos consejos sobre el cuidado y la limpieza de la unidad, para, acto seguido, entrar de lleno en la materia y empezar a proporcionar información muy específica y rigurosa sobre la unidad de *diskettes* y su funcionamiento.

De los tres capítulos principales, el primero está dedicado a la explicación a fondo del funcionamiento de la unidad, sobre todo en sus aspectos mecánicos, llegando a comentar aspectos tan concretos como la función del fieltro de presión, o el funcionamiento de la anilla de paro de la correa del motor. El segundo capítulo está dedicado a la práctica, a los ajustes y comprobaciones. Por ejemplo se enseña, paso a paso, como ajustar la velocidad de giro del *diskette*, o la posición del cabezal de lectura y escritura. Por ultimo, en el tercer capítulo se comentan el diseño y el funcionamiento de la parte electrónica de la unidad. Este capítulo es de lo más completo que hemos visto. Incluye desde una pequeña introducción a la electrónica digital, hasta el esquema completo de la placa, con la disposición de los componentes y los valores de los mismos, pasando por una serie de consejos sobre las mediciones y los ajustes que el usuario podrá realizar para detectar y reparar averías.

El libro se completa con un listado de las piezas de recambio, tanto electrónicas como mecánicas, a emplear en las reparaciones, y por un conjunto de listados de programas que se utilizarán para realizar pruebas de todo tipo sobre la unidad.

Todo el libro viene profusamente ilustrado con esquemas, dibujos, tablas y figuras que servirán de ayuda a la hora de enfrentarse con la **1541**.

LOS MEJORES PROGRAMAS PARA COMMODORE 64

Autores: Robert Erskine, Humphrey Walwyn, Paul Stanley, Michael Bews
Editor: Paraninfo
Paginas: 198
Precio: N.D.

En esta obra se encuentran recopilados 57 juegos, de naturaleza y contenido variado y ameno. Por cada juego se incluye un listado del programa y un pequeño comentario inicial.

Este comentario, en la mayor parte de los programas incluidos, no pasa de ser una sencilla descripción humorística de los objetivos del juego. No obstante, en algunos otros, se da una descripción más detallada de la estructura del programa, indicando la misión de cada grupo de sentencias. No estaría de más ampliar este tipo de descripción a todos los programas presentados. De esa forma, el libro podría ser algo más que un simple compendio de juegos para teclearlos y guardarlos, y el lector interesado podría aprender algo más con los programas.

Los juegos están escritos en BASIC, incluyéndose en algunos pequeñas rutinas en código máquina que se cargan mediante el uso de sentencias READ y DATA.

Se incluyen además en el libro algunos programitas típicos de utilidad, tales como un calendario perpétuo, biorritmos, un generador de carteles publicitarios y un programa de enseñanza de aritmética básica para niños pequeños.

El hecho de que los programas estén escritos en BASIC limita en cierta medida la calidad conseguible a una velocidad dada, pero es muy indicado en un libro de este tipo, para hacer los programas accesibles y no excesivamente tediosos para introducir en la máquina.



TODO SOBRE EL COMMODORE 64

Autor: Craig Chamberlain
Editor: Interamericana
Paginas: 289
Precio: N.D

Un libro que será de extraordinaria utilidad a cualquier programador o usuario poco experimentado, que desee familiarizarse con los conceptos fundamentales del BASIC, así como aprender la forma eficaz de manejo de su **Commodore 64**.

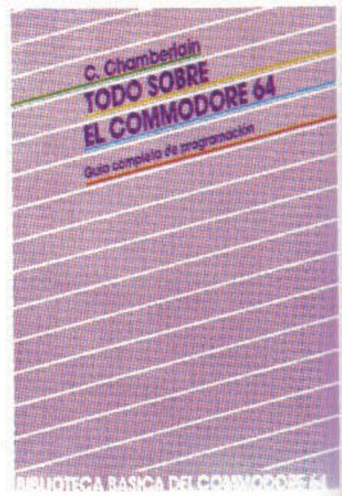
Es por ello que la presentación del libro no es la de un tedioso manual de usuario, con un listado y presentación de cada uno de los comandos e instrucciones. En lugar de eso, se ven cada uno de los conceptos asociados a la programación, explicando claramente su contenido e ilustrando continuamente su uso por medio de ejemplos.

Así, se parte de una introducción preliminar en la que el lector, aunque no disponga de ninguna experiencia previa, introduce dos pequeños programas muy ilustrativos, que despiertan irremediablemente su interés por el resto de la obra.

Se pasa después a desarrollar ideas sobre los elementos de la programación en BASIC, dedicando un capítulo del libro a cada uno de ellos. Se explican aquí las sentencias, las variables, la entrada-salida, las funciones, la lógica condicional, el control, el almacenamiento de datos, las subrutinas y el programa BASIC completo.

Seguidamente se dedica un capítulo al desarrollo de programas. En él se nos ofrecen unas directrices básicas para la especificación, diseño, desarrollo y depuración de nuestros propios programas, acompañadas de la ejecución de un juego como ejemplo final.

Es de resaltar una característica importante a lo largo de todo el libro: siempre que se termina de explicar un apartado o idea, se da un resumen breve de los puntos fundamentales expuestos. Ello hace que



el lector pueda fijar las ideas con mucha mayor claridad.

Por último se incluyen apéndices sobre las teclas de control del cursor y del color, de las tablas de memoria de pantalla y de color, y de los códigos ASCII y de pantalla. Estos apéndices son de uso frecuente para todo programador, experimentado o no.


ocean
ULTIMATE
PLAY THE GAME
U.S. GOLD
All American Software
They sold a
MILLION
They sold a
MILLION
II
BRUCE LEE
PSION
match point
MATCH DAY
Hit SQUAD
ENTOMBED REPLACES KNIGHT LORE ON THE COMMODORE 64
4
GRANDES
EXITOS EN
1
ESTUCHE
DOBLE
TODOS ESTOS
JUEGOS JUNTOS
EN
DOBLE CASSETTE
A PRECIO
DE
UN SOLO
PROGRAMA
The Hit SQUAD
DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA: ERBE SOFTWARE C/ STA. ENGRACIA, 17
28010 MADRID. TEL. (91) 447 34 10 DELEGACION BARCELONA, AVDA. MISTRAL, N.° 10 - TEL. (93) 432 07 31

